NBSIR 77-1383

Interactive Computer Program for the Determination of Reverberation Time

Thomas W. Bartel

Institute for Basic Standards
National Bureau of Standards
Washington, D.C. 20234

December 1977

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NBSIR 77-1383

INTERACTIVE COMPUTER PROGRAM FOR THE DETERMINATION OF REVERBERATION TIME

Thomas W. Bartel

Institute for Basic Standards
National Bureau of Standards
Washington, D.C. 20234

December 1977

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, *Secretary*

Dr. Sidney Harman, *Under Secretary*

Jordan J. Baruch, *Assistant Secretary for Science and Technology*

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Acting Director*

ABSTRACT

A description of the computer program used to measure the reverberation time in a reverberation room is presented. The program controls the operation of a real-time analyzer, a random noise generator, and a microphone multiplexer. The reverberation time for each digitally recorded decay curve is determined from a straight line least-squares fit. The program is written in FORTRAN V and requires approximately 35,000 eight-bit bytes of core memory. Flow charts, source listings, and sample printouts are included.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

I. INTRODUCTION

This report describes the computer program used to measure the reverberation time in a reverberation room according to the Standard Method of Test for Sound Absorption of Acoustical Materials in Reverberation Rooms, ASTM C423-66. The program is written to control the instrumentation system used for the NBS reverberation room. It should, however, be suitable for use in any laboratory with a similar instrumentation system.

In this program, the reverberation decays are digitally recorded simultaneously in the 30 1/3-octave bands from 25 to 20 000 Hz. (The program allows the operator to limit the processing of the data to a narrower frequency range, which may be appropriate for the characteristics of the reverberation room.) Typically 20 s are allowed for sampling the 30 filtered decaying sounds and 15 s for the sound field to reach steady state. (The sound buildup times are typically less than 2.5 s.) During the 15-second buildup the decay rates of the preceding decay are computed from a least-squares curve fit in each of the 1/3-octave bands in the selected frequency range. This procedure permits the decay rates to be recorded and processed as fast as the decays can be physically performed, about 100 decays per hour. The program allows a detailed examination of the decay curves to determine whether or not they can be approximated by single straight lines.

II. INSTRUMENTATION

The program assumes that several elements of the instrumentation system are under computer control. These elements, shown in Fig. 1, are a real-time analyzer, a random noise generator, and a microphone multiplexer. The real-time analyzer filters the input signal through 30 contiguous

1/3-octave band-pass filter channels, samples each channel simultaneously for an integration period, determines the rms level for each channel, and transmits the digital results to the computer. For the NBS system, an integration time of 0.1 s is used for reverberation time measurements. The time required to transmit the digitized output level from all 30 1/3-octave filters to the computer is approximately 1 ms.

The output of the random noise generator is turned on and off by the computer. For most measurements made with the NBS system, unfiltered pink noise is sent to the amplifier and loudspeaker, resulting in sufficient dynamic range to measure the reverberation time simultaneously in the 1/3-octave bands from 80 Hz to 10 kHz.

The computer controls the channel selection of a microphone multiplexer. In the NBS system, the multiplexer has 16 channels, of which 12 are used with the 12-microphone array installed in the reverberation room. In the reverberation time program, the multiplexer is used, at the operator's option, to select the next microphone channel after each decay.

III. DIGITAL PROCEDURE TO DETERMINE THE REVERBERATION TIME

The reverberation time for each decay curve is determined from a straight line least-squares fit to the digitized output of each 1/3-octave filter in the following manner. First the digitized output of each filter is converted by the real-time analyzer to an rms value using an averaging time of 0.1 s. This rms value for each filter is read by the computer in less than 1.0 ms and another 0.1-second averaged rms value is obtained, and so on. These rms values are depicted by the solid circles in Fig. 2.

Before the actual decay curve can be defined two steady-state values for each 1/3-octave band are determined, the long-term average signal level prior to the abrupt cessation of the sound in the room and the long-term average system noise level (both acoustical and electrical) of each microphone. The average system noise level in each 1/3-octave band is determined by sampling the signal when the sound source is turned off. This is denoted the average noise level in Fig. 2. The averaging time, determined by the number of consecutive 0.1-s samples taken, is chosen by the operator. These noise levels are assumed to remain constant throughout the entire set of measurements. The long-term average signal level, denoted the average signal level in Fig. 2, is obtained just prior to each decay by sampling the signal for a short time before the sound source is turned off. The operator chooses the number of consecutive 0.1-s samples taken before and after the sound cutoff (maximum total time of 20 s). Typical values are 4 s and 16 s, respectively.

Since the point at which each decay curve starts is often poorly defined, the upper limit for the least-squares curve fitting procedure is arbitrarily set x dB below the average signal level. A typical value for x is 5 dB. Similarly the effects of the system noise must not influence the least squares fit and, therefore, the lower limit is arbitrarily set y dB above the average noise level. A typical value for y is 10 dB. Both x and y are chosen by the operator. The limits for the curve fitting procedure are denoted the upper and lower amplitude limits in Fig. 2.

Since the distribution of the 0.1-second averaged rms values about the fitted line defining the decay curve can be relatively large, an iterative procedure is used to determine the first and last data points to

be used by the least squares procedure. The initial decay curve is determined from a least squares fit starting with the first point after the sound has been turned off and ending with the first point to reach the noise level. The intersection of this initial decay curve with the upper and lower amplitude limits is then determined. The data points nearest these intersections are chosen as the new upper and lower points for the curve fitting procedure and a new decay curve is determined. This process is repeated until the intersections of the upper and lower amplitude limits with the decay curve are within 0.1 s of the points previously used to determine the decay curve. This procedure generally requires fewer than five iterations and the decay times for all 22 1/3-octave bands from 80 and 10 000 Hz can be determined in less than 15 s.

When one wants to determine if a decay curve contains a "break point" resulting from two distinct decay times, the program may first be run with the value of y increased such that the decay curve is computed using data that only span 15 or 20 dB. Then the program may be rerun with the value of x increased to 15 or 20 dB and the value of y decreased to its original value. If these two measurements result in statistically distinguishable reverberation times in any 1/3-octave band, then the room has two distinct decay times in that band.

IV. PROGRAM SEQUENCE

The manner in which the reverberation time measurement is performed is summarized in the abbreviated flowchart given in Appendix A. At the start of the program the operator enters the desired confidence interval and level of the average reverberation times, the 1/3-octave frequency range, and the time between samples, which is approximately 0.1 s but is accurately determined in advance with a time-interval counter.

The average signal and ambient noise levels are then determined. This section of the program enables the operator to adjust the loudspeaker amplifier, microphone and real-time analyzer gains to achieve maximum dynamic range while remaining below the signal level overload region of the real-time analyzer (see Printout 1, Appendix C). Note that the signal and noise levels in the printouts are the averages of several successive 0.1-s samples obtained from the real-time analyzer. In the NBS system, the resolution of the real-time analyzer is 0.25 dB. The levels are printed to the nearest 0.01 dB in order to aid in tracing the computations performed on the data, rather than to represent the uncertainty of the data.

To allow for different noise levels among the microphone channels used in the measurement, the operator may conduct a separate ambient noise measurement for each channel (see Printout 2, Appendix C). After these adjustments the ambient noise levels are obtained and stored for determining the lower curve fitting limits of the decay curves.

The operator then enters the number of samples to be taken before and after the sound is cut off, and the values x and y, discussed in the previous section, to be used in determining the curve-fitting limits. One decay is sampled upon initiation by the operator, who may observe the curve-fitting parameters (see Printout 1, Appendix D) to ensure that the measurement is proceeding correctly. Following this the operator makes a decision of "YES," "NO," or "AUTO" in response to the question "Do you want to conduct another decay?" The results of these responses are as follows:

YES: Conduct and analyze one more decay. Operator controls the microphone selection and initiates the sampling of the decay.

AUTO: Conduct and analyze several decays automatically. Operator first chooses the number of decays to be taken, then the operation of the microphone multiplexer (step or not step between decays), and finally the insertion of any delay time for sound buildup that is desired in addition to the time required to compute the reverberation times, which is a maximum of 15 s but shorter if a narrower frequency range has been chosen.

NO: Do not conduct any more decays at this point.

If the operator entered "NO" to the above question, he may then examine the decay curves of the last decay to inspect their linearity and the superposition of the computed least-squares fit. Such a display is shown in Printouts 1 and 2, Appendix E. Printout 1 displays, in tabular form, the level transmitted by the real-time analyzer for each 0.1-s integration period for a specified 1/3-octave band. The levels are printed in horizontal rows of 10. Printout 2 is a graphical display of the same data. Following this, he may request a display of the average reverberation time for each 1/3-octave frequency band of all of the decays conducted up to this point (see Printout 1, Appendix F). Also displayed are the confidence intervals achieved at this point, and the estimated number of decays remaining in each band to attain the confidence interval specified at the beginning of the measurement.

The operator may then choose to conduct more decays using the same "YES," "NO," or "AUTO" choices described previously. If "NO" is entered, the measurement is terminated following a printout of the results (see Printouts 1 and 2, Appendix G).

V. SOFTWARE DETAILS

The programming language used is the Interdata FORTRAN V Level 1 Software System.* It is a superset of ANSI Standard X3.9-1966 FORTRAN and should be compatible or nearly compatible with the FORTRAN V used on other computer systems. Users of this program on other systems should check the compatibility of FORMAT statements using the A Format and INTEGER*2 statements which define the storage size of integers to be two 8-bit bytes.

The program for measuring reverberation time consists of a main program, REVERB, written in FORTRAN V, several FORTRAN subroutines, and several assembly language subroutines written for use on an Interdata Model 70 minicomputer. These elements are listed in Table 1. The corresponding flow charts, sample printouts, and listings are given in the appendices. The last appendix, Appendix N, gives the listings of a separate FORTRAN program, GENI, and three of its subprograms, STUDIN, FISHIN, and FISH, that are not a part of the program REVERB. They are used to generate the t-distribution table read from logical unit 4 by the function subprogram STUDIM, called by REVERB.

---

*The commercial computer products utilized are identified in order to adequately describe the program elements discussed in this report. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that these products are necessarily the best available for the purpose.

In the NBS system, the subroutines SIGNAL, CURVFT, DSDATA, AVRGE, and RSULTS are loaded as overlays called by the main program, REVERB. Each call to these subroutines is therefore preceded by the instructions REWIND 0 and CALL IFETCH, which cause the overlay to be loaded. In addition, there is a branch to "End of Job" if there is an overlay load error. These instructions should be deleted or bypassed if this program is used in a system not requiring overlays. The subroutines will then be loaded and called as regular subroutines.

In the NBS system the complete task occupies 35,452 eight-bit bytes of memory, of which 12,000 bytes are used for storing the real-time analyzer data for each decay (200 samples/1/3-octave band x 30 bands = 6000 samples, with each sample being stored as a 2-byte integer).

The flow charts, given in the appendices, conform to the American National Standard Flowchart Symbols and Their Usage in Information Processing, X3.5-1970. In these flow charts, the small numbers enclosed in parentheses refer to FORTRAN statement labels in the program listings.

In the program listings, the first three statements of every program or subprogram are control statements used to insert the program name, in the form of a binary label, into the compiled object code.

VI. INPUT/OUTPUT LOGICAL UNITS

The following logical units are used for input/output operations:

LU0 - File from which overlays are loaded (see Section V).

LU1 - Scratch file for accumulating a table of the reverberation times for each decay; this file is read when computing the average reverberation times.

LU2 - Output file for storing the final table of averaged reverberation times; can be used as input to other programs, such as the sound power program.

LU3 - Hard copy printout device. Should be assigned to a 132 character line printer. During execution of the program, the operator may choose the printout device logical unit to obtain either a hard copy printout or a view of the results on the CRT terminal.

LU4 - File containing a table of the t-distribution for the chosen confidence level. This table is used by the function subprogram STUDIM, and must be created by another program, GEN1, given in Appendix N. This file must be a random record access file.

LU5 - Operator communication device. Should be assigned to a CRT terminal. Through this device the operator controls the course of the measurement, making the decisions that are called for and entering certain measurement parameters, which include:

1. The desired confidence interval and confidence level.
2. The desired 1/3-octave frequency bands.
3. The number of samples to be taken for each decay, both before and after the sound source is cutoff.
4. The upper and lower curve-fitting limits.
5. The number of decays to be conducted automatically.
6. Whether or not to step the microphone multiplexer between decays.

Transmission of data to and from the real-time analyzer, microphone multiplexer, and random noise generator is not handled through logical units, but through separate subroutines written in the assembly language for the Interdata Model 70 minicomputer. These subroutines are included in TABLE 1; their listings are given in Appendix M.

TABLE 1. REVERBERATION TIME PROGRAM ELEMENTS

PROGRAM NAME	TYPE OF PROGRAM	FUNCTION	SUBROUTINES CALLED
REVERB	FORTRAN Program	Performs reverberation room decay measurements and computes reverberation time.	SIGNAL, CURVFT, DSDATA, AVRGE, RSULTS, SAMPLE, SOUND, DELAY, MUX, BAND, RTA, HUMID, STUDIM, TINORM, SUBMUX
SIGNAL	FORTRAN Subroutine	Determines signal-to-noise levels. Allows operator to adjust speaker amplifier gain and measurement system gains to achieve optimum signal-to-noise levels before beginning decay measurements.	SAMPLE, BAND, SOUND, SUBMUX, MUX, DELAY
CURVFT	FORTRAN Subroutine	Performs least-squares curve fit on decay data.	BAND
DSDATA	FORTRAN Subroutine	Displays real-time analyzer data for one decay, in both tabular form and as a graphic plot. The plot also shows the least-squares curve fit superimposed on the data points.	BAND
AVRGE	FORTRAN	Averages the results of many decays. Prints a table indicating, band by band, whether the desired confidence interval has been achieved. For those bands that do not achieve it, the approximate number of decays still necessary to achieve the desired confidence interval is indicated.	TINORM, STUDIM
RSULTS	FORTRAN Subroutine	Displays the results after final decay has been taken, which includes graphic plot of total room absorption versus frequency.	HUMID

TABLE 1. (Continued)

PROGRAM NAME	TYPE OF PROGRAM	FUNCTION	SUBROUTINE CALLED
HUMID	FORTRAN Subroutine	Computes relative humidity from wet and dry bulb temperatures. Also computes speed of sound.	None
SAMPLE	FORTRAN Subroutine	Takes a specified number of real-time analyzer samples and checks for signal levels that exceed its maximum input level.	RTA
BAND	FORTRAN Subroutine	Converts one band of real-time analyzer samples from fixed-point to floating-point numbers.	None
SUBMUX	FORTRAN Subroutine	Steps the microphone multiplexer to a given microphone channel.	MUX
TINORM	FORTRAN Function Subprogram	Computes value of the inverse of the standard normal distribution function for a specified probability.	None
STUDIM	FORTRAN Function Subprogram	Finds value of inverse of student's t-distribution for a specified argument by reading from a table generated on a storage file by the program GENI	None
MUX	Assembly Language Subroutine, FORTRAN Callable	Steps and reads multiplexer	None
RTA	Assembly Language Subroutine, FORTRAN Callable	Controls and reads real-time analyzer. Also shuts off sound after the specified number of samples have been taken.	None
SOUND	Assembly Language Subroutine, FORTRAN Callable	Turns random noise generator on and off.	None
DELAY	Assembly Language Subroutine, FORTRAN Callable	Provides a delay of 0 to 32.767 s in approximate 1-ms increments.	None

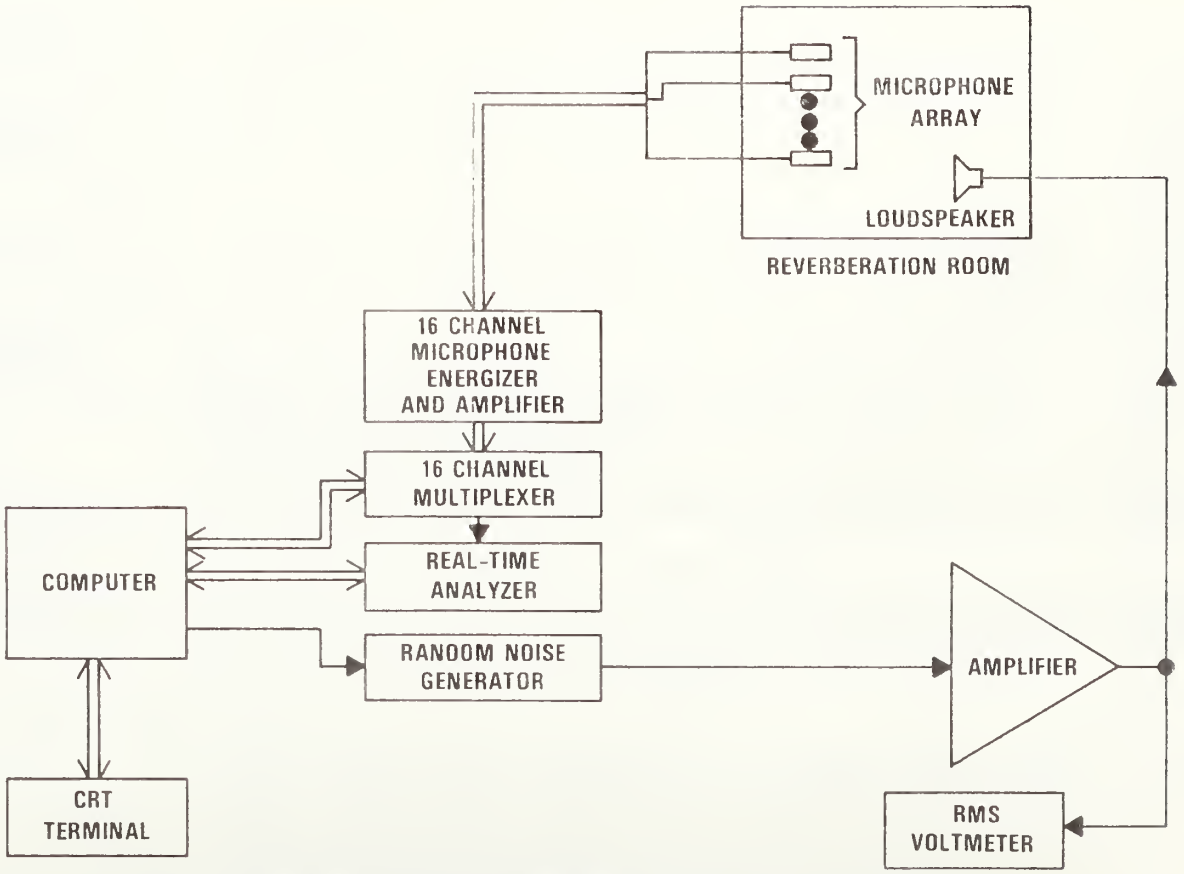


FIGURE 1. SIMPLIFIED SCHEMATIC OF INSTRUMENTATION SYSTEM USED TO DETERMINE THE REVERBERATION TIME.

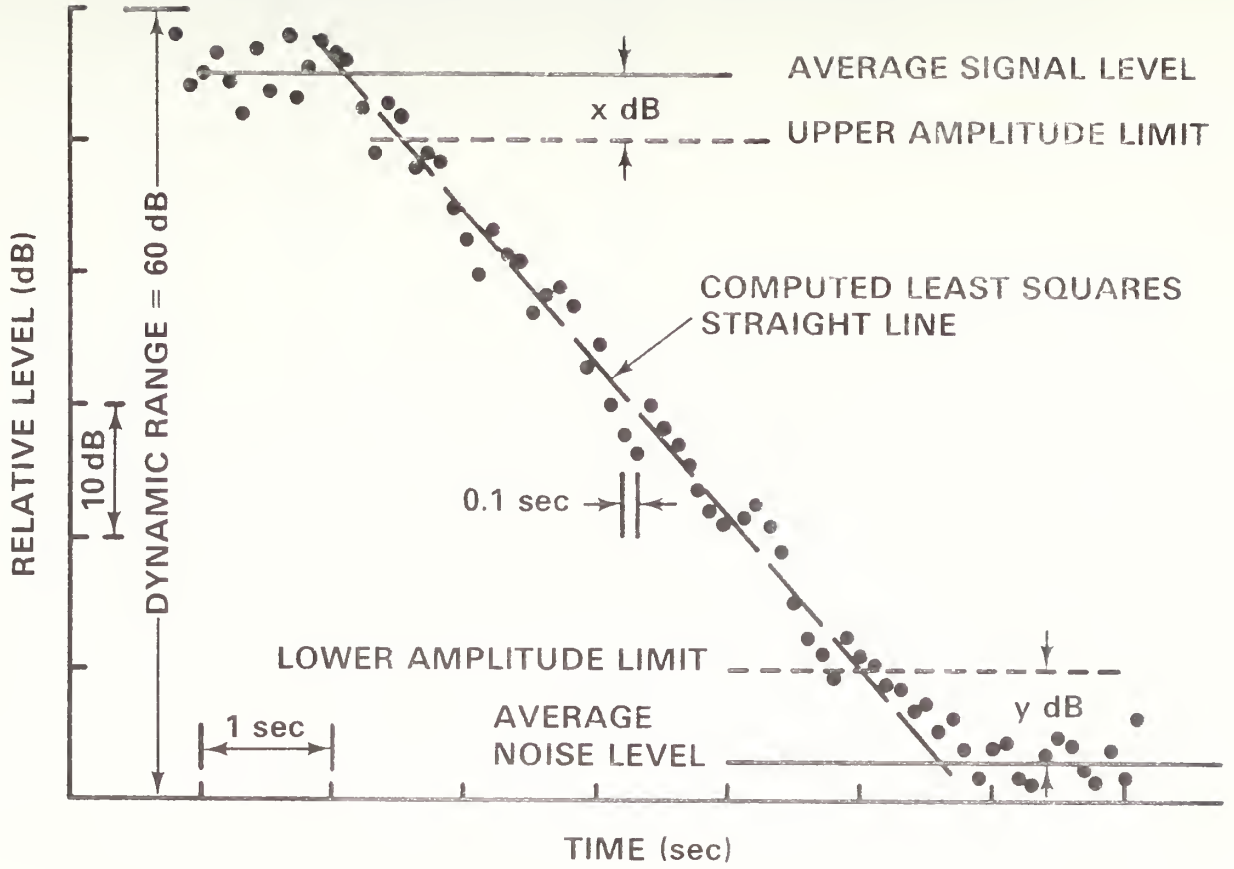
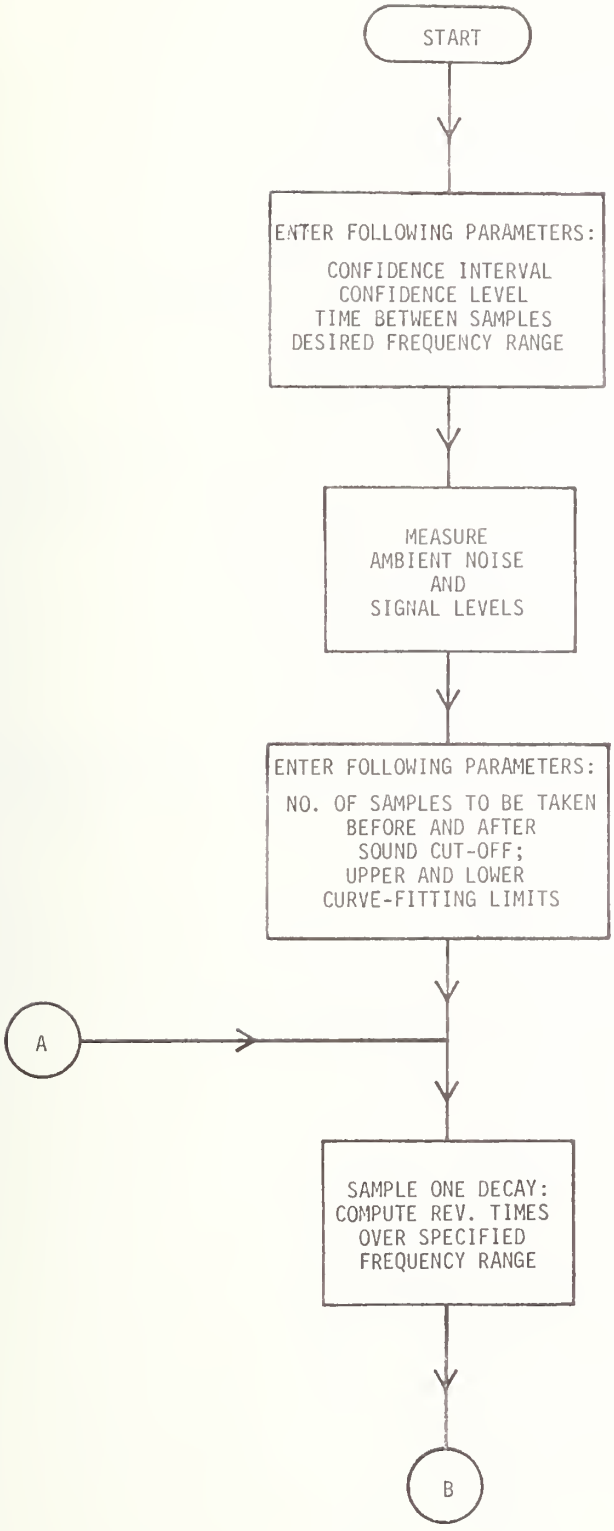


FIGURE 2. TYPICAL PLOT OF A DIGITALLY-OBTAINED DECAY CURVE AND THE DEFINITIONS USED TO DETERMINE THE REVERBERATION TIME.

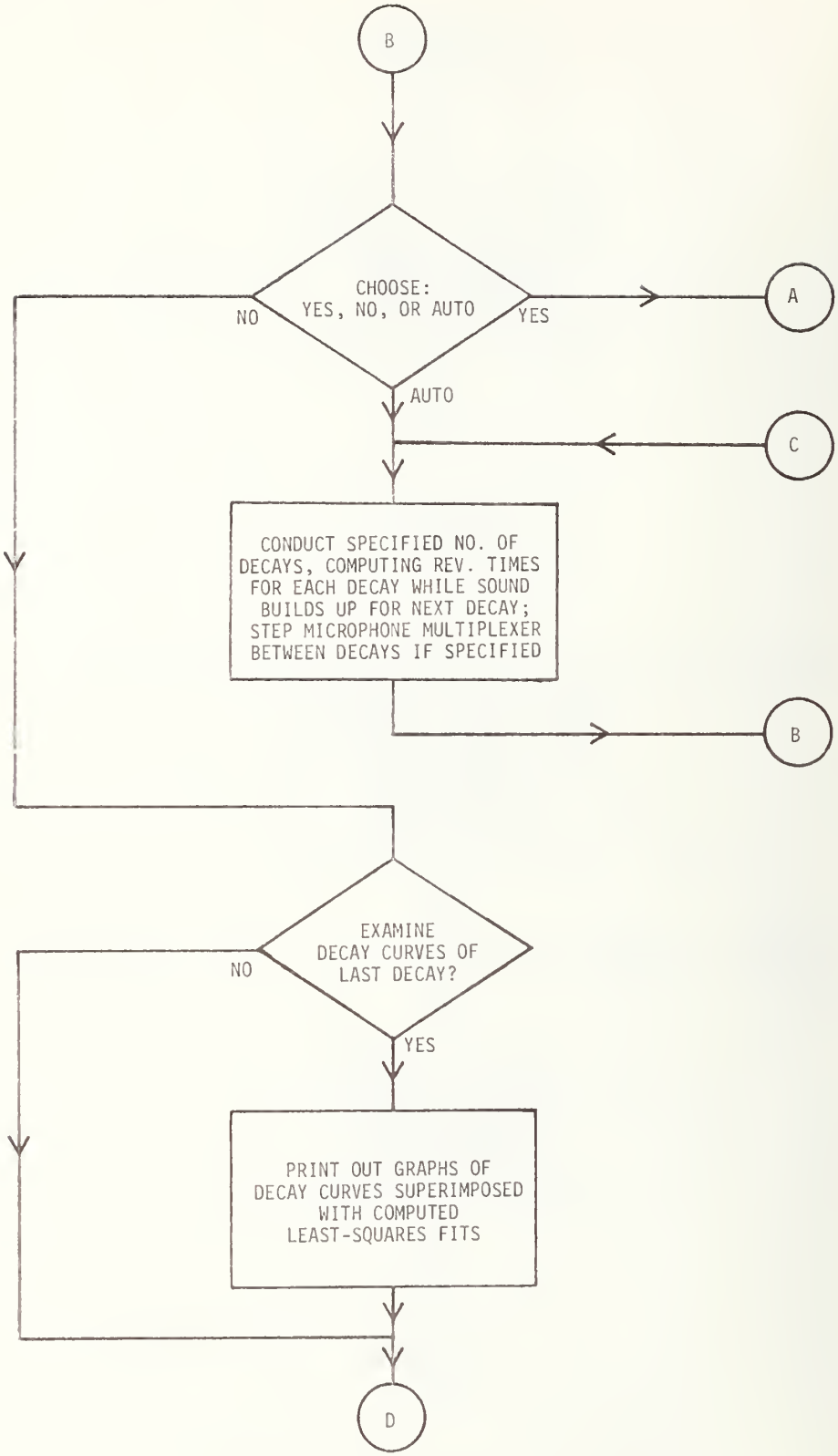
APPENDIX A

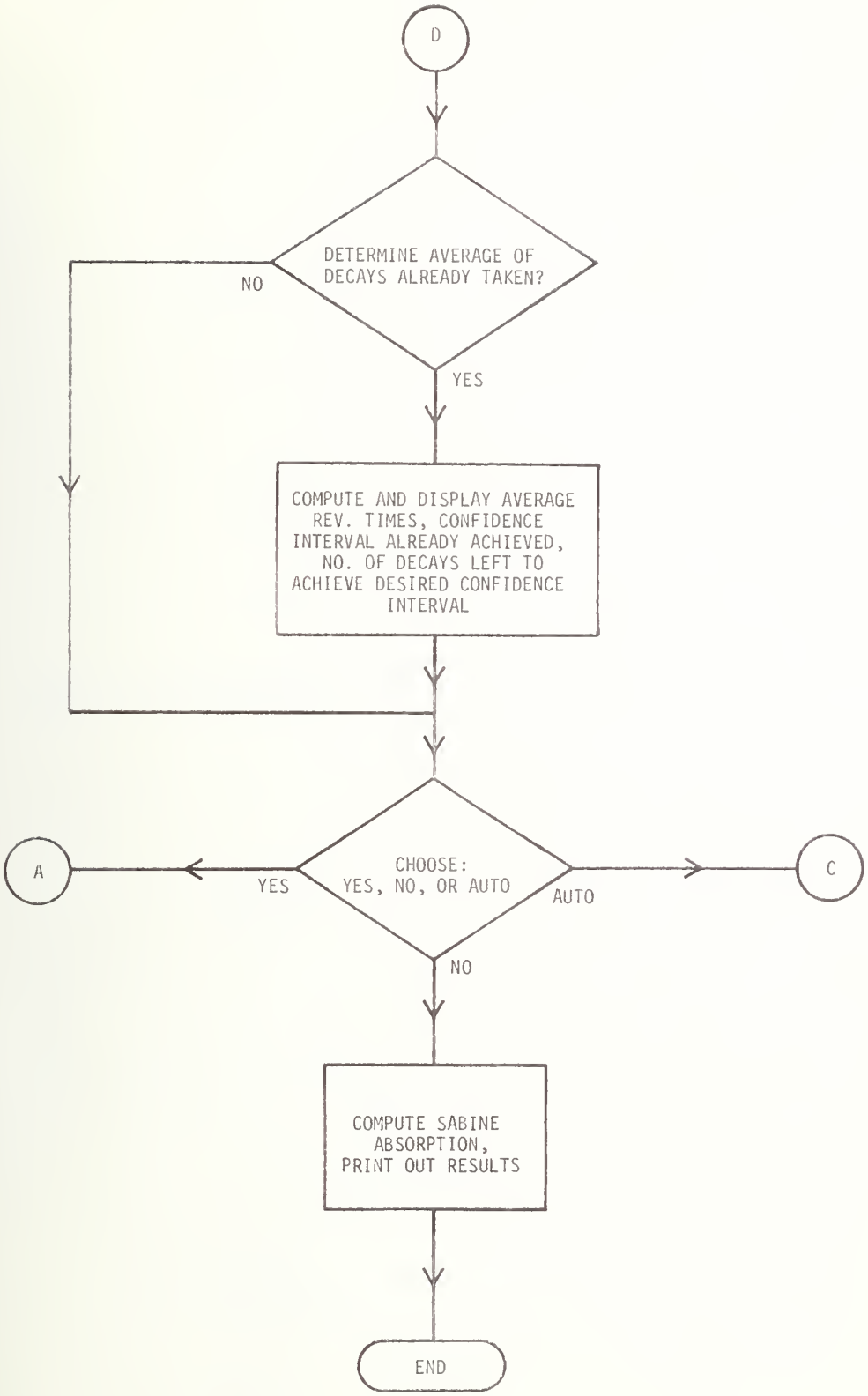
Abbreviated Flow Chart of Reverberation Time Program

ABBREVIATED FLOW CHART: REVERB



ABBREVIATED REVERB-2

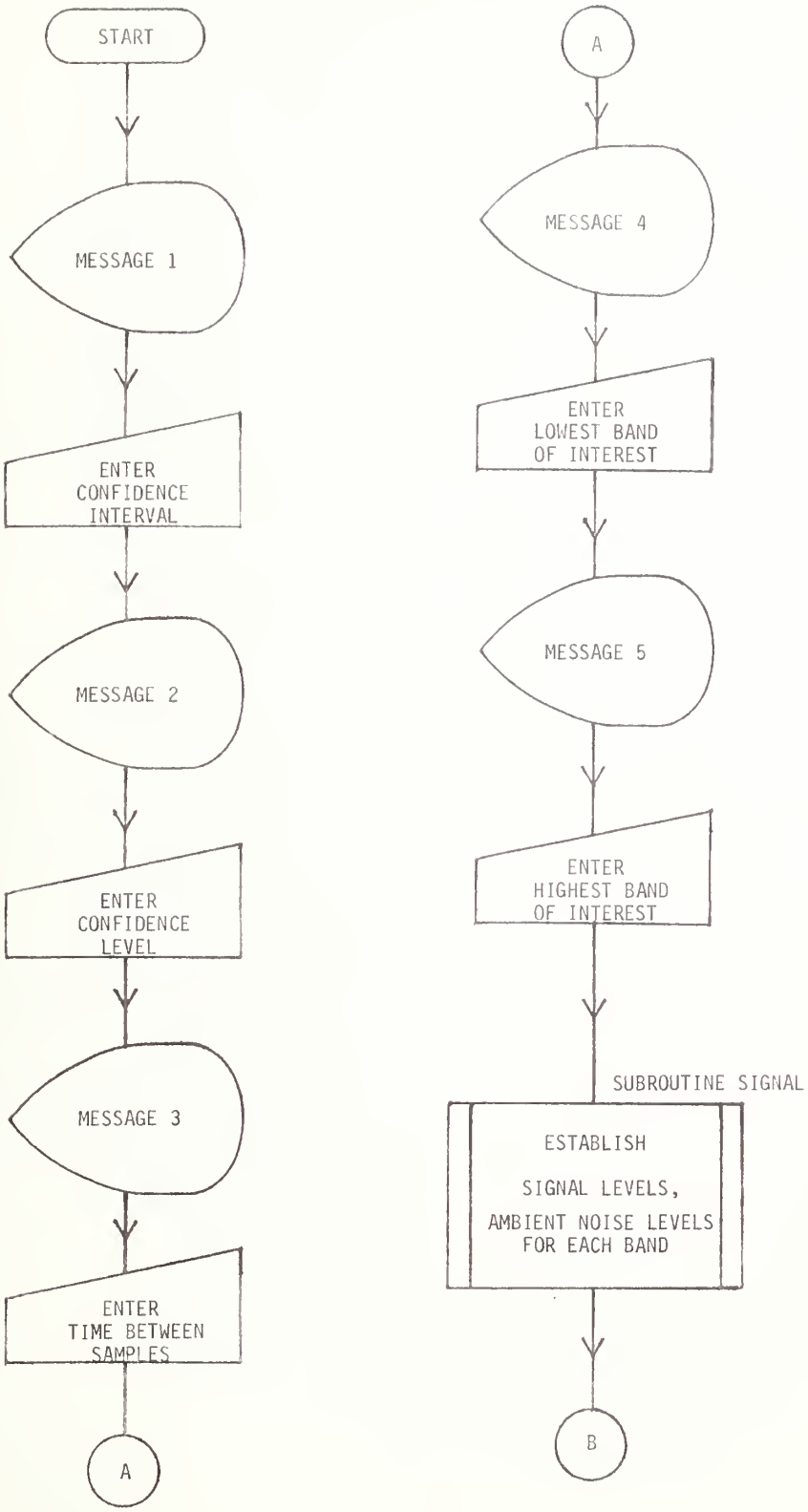


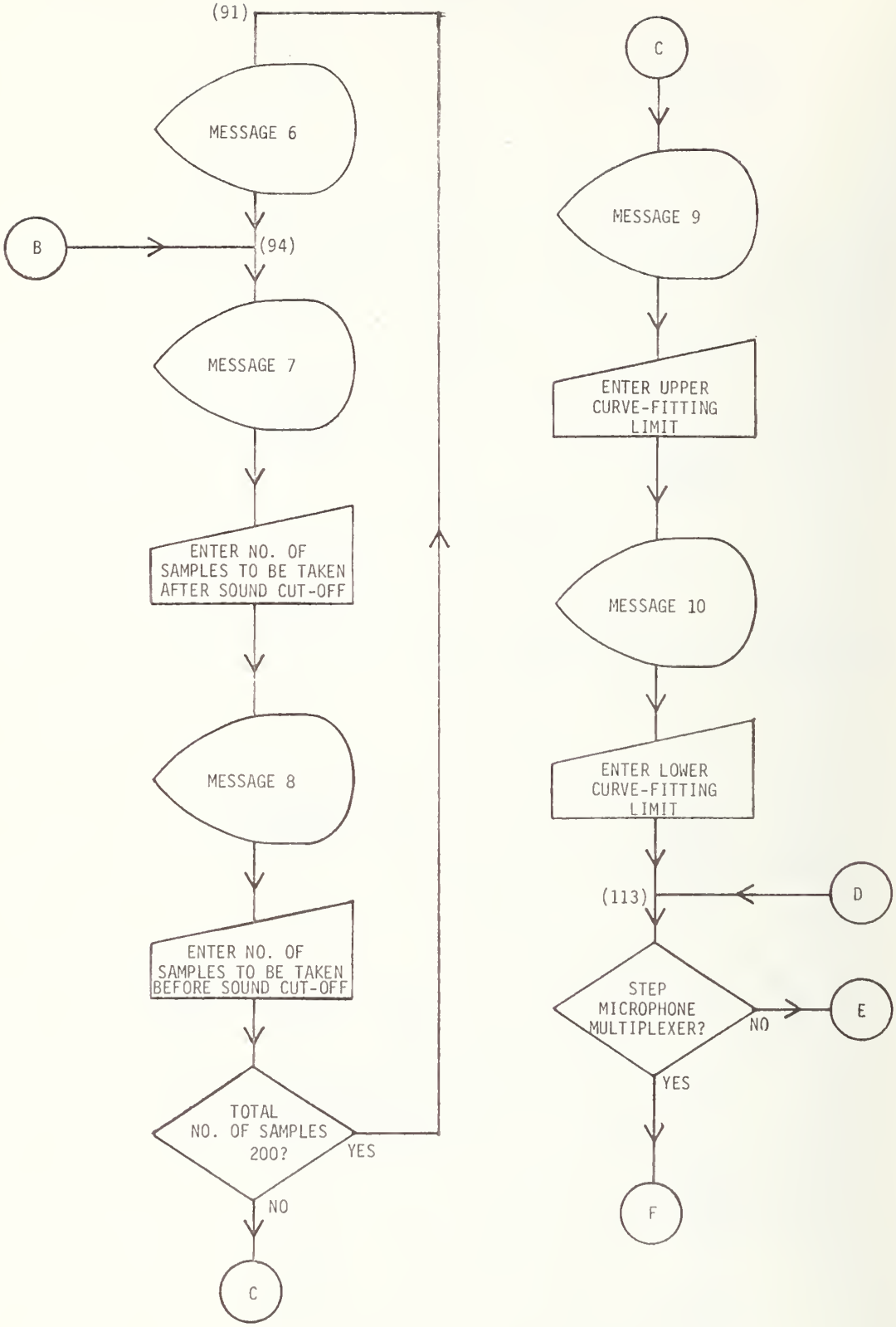

D
DETERMINE AVERAGE OF DECAYS ALREADY TAKEN?
NO
YES
COMPUTE AND DISPLAY AVERAGE REV. TIMES, CONFIDENCE INTERVAL ALREADY ACHIEVED, NO. OF DECAYS LEFT TO ACHIEVE DESIRED CONFIDENCE INTERVAL
A
CHOOSE: YES, NO, OR AUTO
YES
AUTO
C
NO
COMPUTE SABINE ABSORPTION, PRINT OUT RESULTS
END

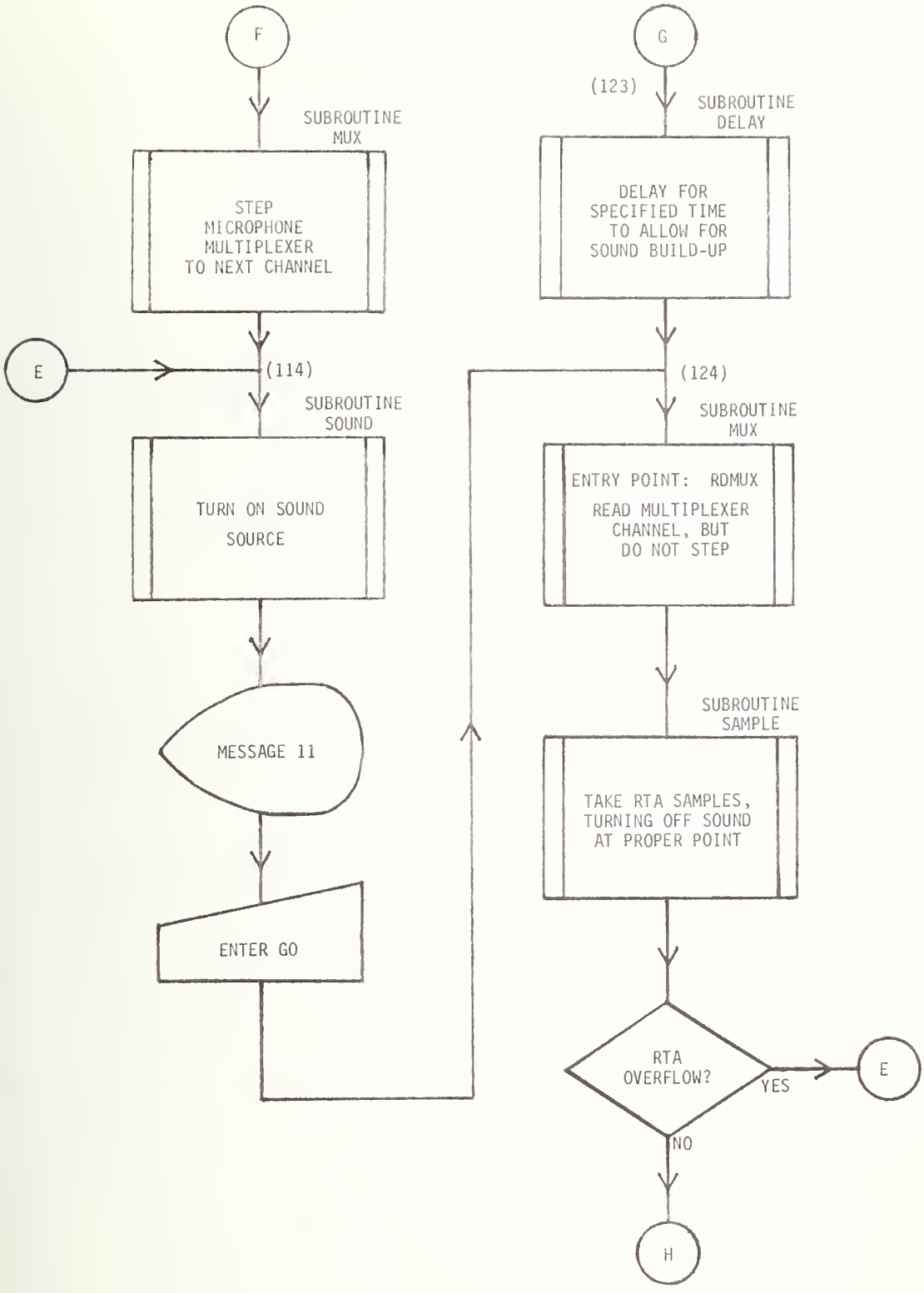
APPENDIX B

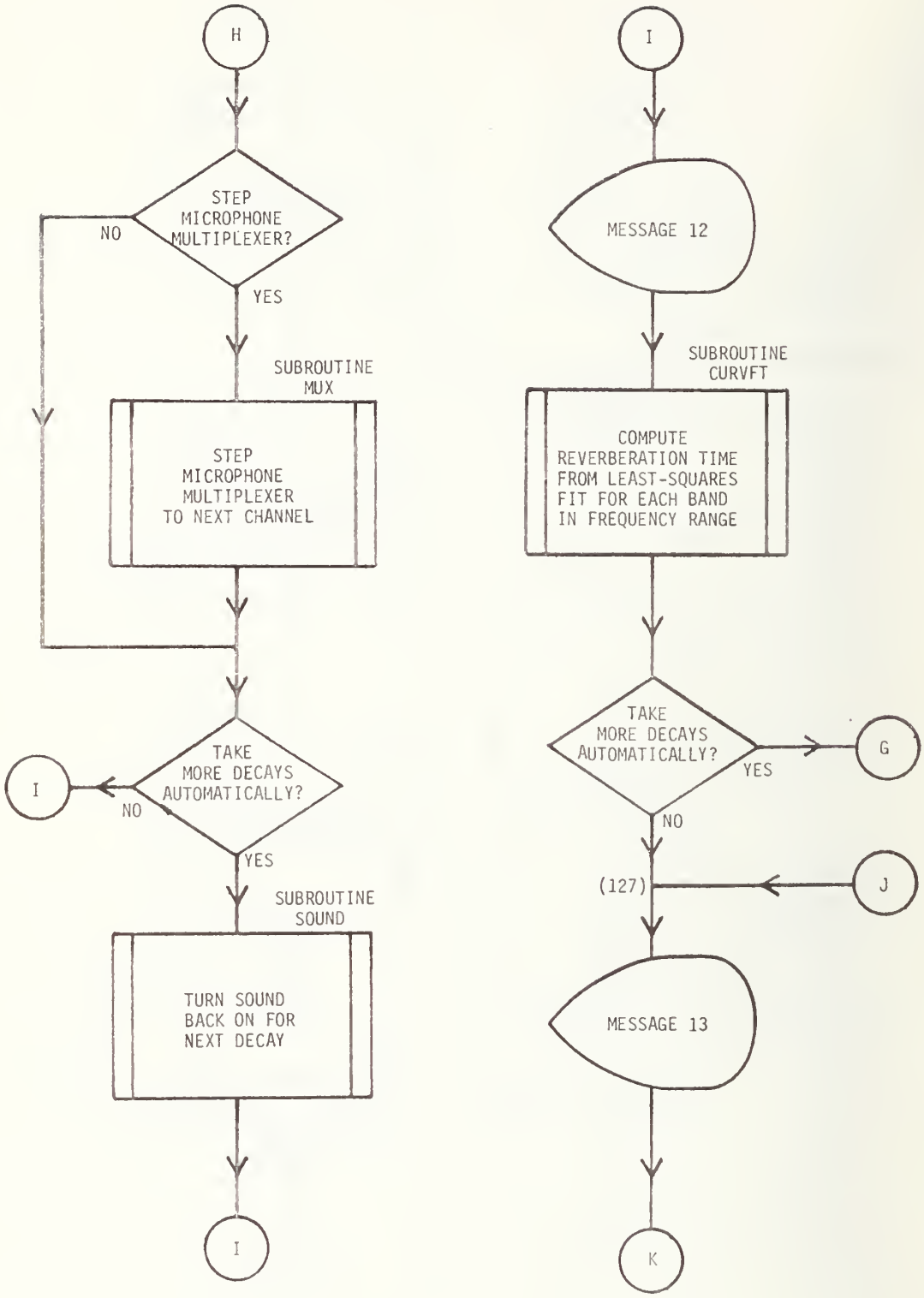
Program REVERB Flow Chart, Terminal Messages, and Listings

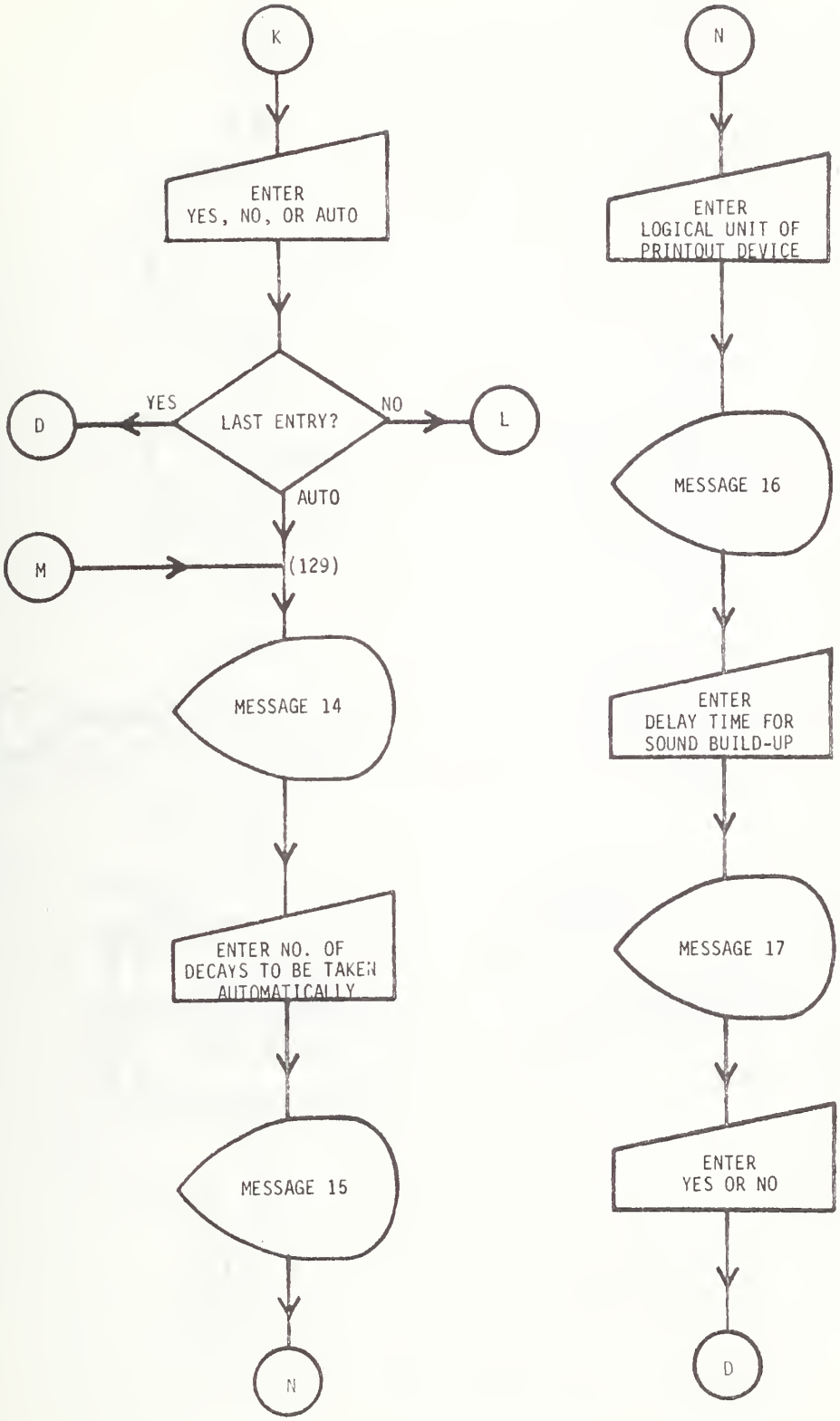
FLOW CHART: REVERB

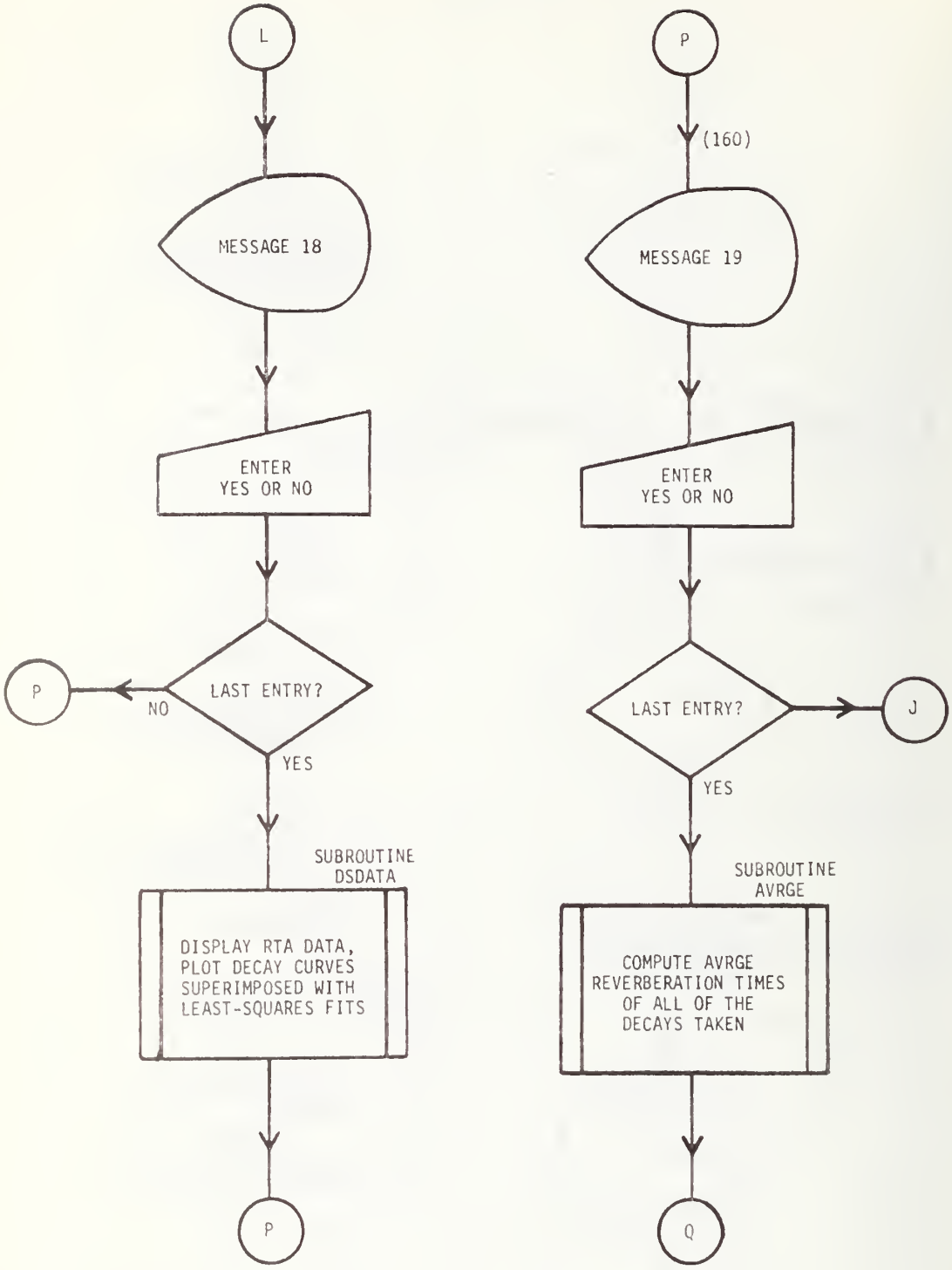


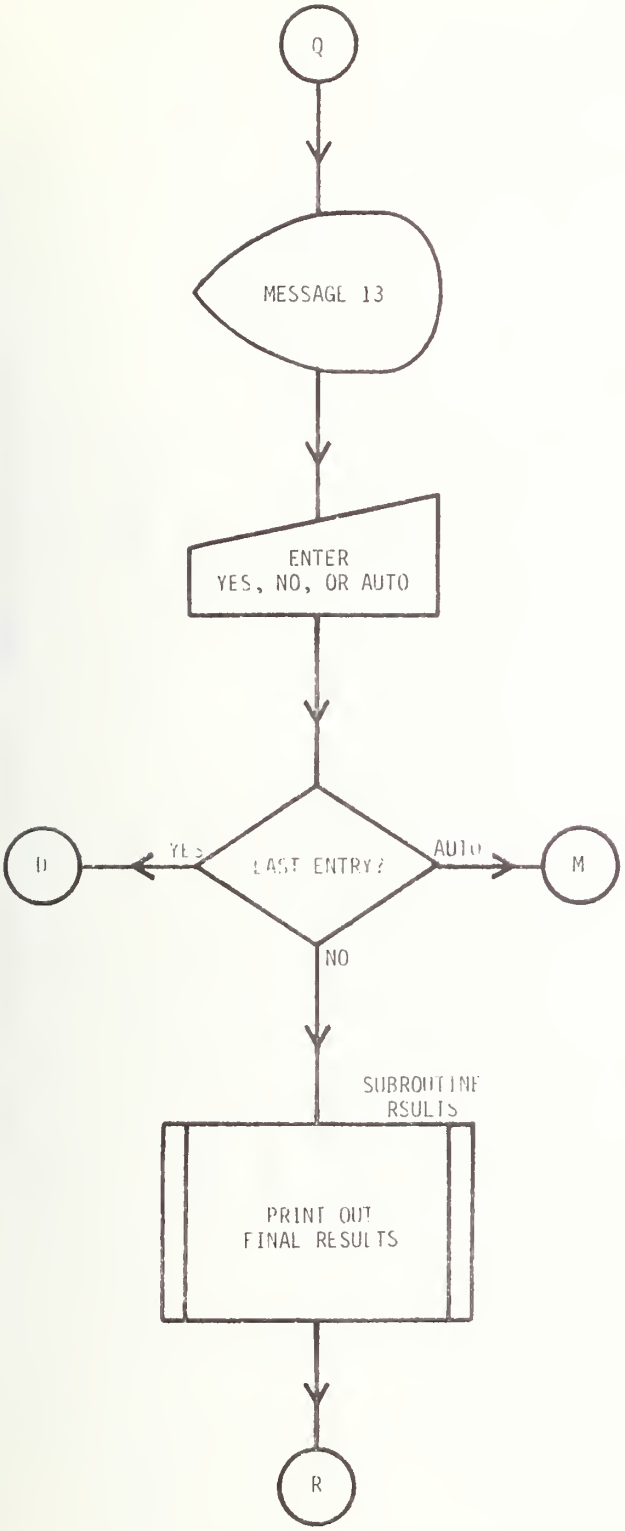

(91)
MESSAGE 6
B
(94)
MESSAGE 7
ENTER NO. OF SAMPLES TO BE TAKEN AFTER SOUND CUT-OFF
MESSAGE 8
ENTER NO. OF SAMPLES TO BE TAKEN BEFORE SOUND CUT-OFF
TOTAL NO. OF SAMPLES 200?
YES
NO
C
C
MESSAGE 9
ENTER UPPER CURVE-FITTING LIMIT
MESSAGE 10
ENTER LOWER CURVE-FITTING LIMIT
(113)
D
STEP MICROPHONE MULTIPLEXER?
NO
E
YES
F

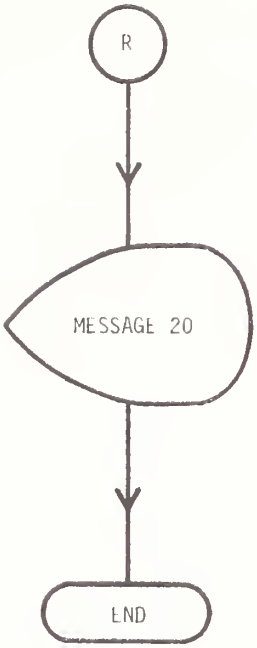

F
SUBROUTINE
MUX
STEP
MICROPHONE
MULTIPLEXER
TO NEXT CHANNEL
E
(114)
SUBROUTINE
SOUND
TURN ON SOUND
SOURCE
MESSAGE 11
ENTER GO
G
(123)
SUBROUTINE
DELAY
DELAY FOR
SPECIFIED TIME
TO ALLOW FOR
SOUND BUILD-UP
(124)
SUBROUTINE
MUX
ENTRY POINT: RDMUX
READ MULTIPLEXER
CHANNEL, BUT
DO NOT STEP
SUBROUTINE
SAMPLE
TAKE RTA SAMPLES,
TURNING OFF SOUND
AT PROPER POINT
RTA
OVERFLOW?
YES
E
NO
H


H
STEP MICROPHONE MULTIPLEXER?
NO
YES
SUBROUTINE MUX
STEP MICROPHONE MULTIPLEXER TO NEXT CHANNEL
TAKE MORE DECAYS AUTOMATICALLY?
I
NO
YES
SUBROUTINE SOUND
TURN SOUND BACK ON FOR NEXT DECAY
I
I
MESSAGE 12
SUBROUTINE CURVFT
COMPUTE REVERBERATION TIME FROM LEAST-SQUARES FIT FOR EACH BAND IN FREQUENCY RANGE
TAKE MORE DECAYS AUTOMATICALLY?
YES
G
NO
(127)
J
MESSAGE 13
K


K
ENTER
YES, NO, OR AUTO
YES
LAST ENTRY?
NO
D
L
AUTO
M
(129)
MESSAGE 14
ENTER NO. OF
DECAYS TO BE TAKEN
AUTOMATICALLY
MESSAGE 15
N
N
ENTER
LOGICAL UNIT OF
PRINTOUT DEVICE
MESSAGE 16
ENTER
DELAY TIME FOR
SOUND BUILD-UP
MESSAGE 17
ENTER
YES OR NO
D


L
MESSAGE 18
ENTER
YES OR NO
P
NO
LAST ENTRY?
YES
SUBROUTINE
DSDATA
DISPLAY RTA DATA,
PLOT DECAY CURVES
SUPERIMPOSED WITH
LEAST-SQUARES FITS
P
P
(160)
MESSAGE 19
ENTER
YES OR NO
LAST ENTRY?
J
YES
SUBROUTINE
AVRGE
COMPUTE AVRGE
REVERBERATION TIMES
OF ALL OF THE
DECAYS TAKEN
Q


Q
MESSAGE 13
ENTER
YES, NO, OR AUTO
D
YES
LAST ENTRY?
AUTO
M
NO
SUBROUTINE
RSULTS
PRINT OUT
FINAL RESULTS
R


R
MESSAGE 20
END

CRT TERMINAL MESSAGES: PROGRAM REVERB

MESSAGE NO.	FORMAT NO.	MESSAGE
1	10	THIS PROGRAM RUNS AND ANALYZES REV ROOM DECAY MEASUREMENTS THE FOLLOWING LOGICAL UNIT ASSIGNMENTS ARE REQUIRED: LU0 -- FILE FROM WHICH OVERLAYS ARE LOADED LU1 -- SCRATCH FILE LU2 -- STORAGE FILE FOR STORING REVERBERATION TIME DATA LU3 -- HIGH SPEED PRINTER LU4 -- STORAGE FILE CONTAINING TABLE OF T-DISTRIBUTION LU5 -- CRT TERMINAL ENTER THE DESIRED CONFIDENCE INTERVAL TO BE ATTAINED IN TERMS OF PERCENT OF THE MEAN REV. TIME (X.X)
2	30	ENTER THE DESIRED CONFIDENCE LEVEL IN PERCENT (XX.XX)
3	42	ENTER THE TIME BETWEEN SAMPLES IN SECONDS (X.XXXXX)
4	45	ENTER LOWEST BAND OF INTEREST (MIN=14)
5	55	ENTER HIGHEST BAND OF INTEREST (MAX=43)
6	92	YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF 200 SAMPLES
7	95	ENTER NO. OF SAMPLES REQUIRED FOR DECAY MEASUREMENT (3 DIGITS)
8	100	ENTER NO. OF SAMPLES TO BE TAKEN BEFORE SOUND CUT-OFF (3 DIGITS)

CRT TERMINAL MESSAGES: PROGRAM REVERB (page 2)

MESSAGE NO.	FORMAT NO.	MESSAGE
9	105	ENTER UPPER CURVE FITTING LIMIT IN DB DOWN FROM SIGNAL LEVEL (NORMALLY 5.0)
10	106	ENTER LOWER CURVE FITTING LIMIT IN DB UP FROM NOISE LEVEL (NORMALLY 10.0)
11	115	THE SOUND HAS BEEN TURNED ON ENTER "GO" TO START DECAY MEASUREMENT
12	125	DECAY NO. 1
13	128	DO YOU WANT TO CONDUCT ANOTHER DECAY? (ENTER "AUTO" FOR AUTOMATIC MODE)
14	130	ENTER NUMBER OF DECAYS TO BE CONDUCTED AUTOMATICALLY (4 DIGITS)
15	133	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE FOR CURVE-FITTING PARAMETERS
16	136	ENTER NO. OF SECONDS OF DELAY REQUIRED FOR SOUND BUILD-UP (XX.X) (MAX = 32) (THIS DELAY IS IN ADDITION TO THE COMPUTING AND PRINTOUT TIME)
17	138	DO YOU WANT TO STEP THE MICROPHONE MULTIPLEXER BEFORE EACH DECAY?
18	150	DO YOU WANT TO SEE THE DATA POINTS OF THE LAST DECAY?
19	170	DO YOU WANT TO SEE THE AVERAGES?
20	230	END OF JOB

PROGRAM REVERB

```
$ASSM
REVERB PROG REVERB - PROGRAM TO MEASURE REVERBERATION TIME
$FORT
C   ------------------------------
C   OVERLAYS CALLED:      SIGNAL,CURVFT,DSDATA,AVRGE,RSULTS
C   SUBPROGRAMS CALLED:   SAMPLE,SOUND,DELAY,MUX,
C         [INDIRECTLY]:   HUMID,BAND,RTA,STUDIM,TINORM,SUBMUX
C   ------------------------------
      INTEGER*2 B,NSIG,N,X0(6060),FREQ(30),LINE(121),
     1BLOW,BHIGH,OVFLOW,AUTO,MM1(30),MM2(30)
      DIMENSION X(200),S0(30,12),S1(30),ARRAY1(30),ARRAY2(30)
      DATA FREQ/25,32,40,50,63,80,100,125,160,200,250,315,400,500,
     1630,800,1000,1250,1600,2000,2500,3150,4000,5000,6300,8000,
     210000,12500,16000,20000/
      WRITE (5,10)
10    FORMAT (//'THIS PROGRAM RUNS AND ANALYZES REV ROOM DECAY
     1 MEASUREMENTS'//
     2'THE FOLLOWING LOGICAL UNIT ASSIGNMENTS ARE REQUIRED:'/
     35X,'LU0 -- FILE FROM WHICH OVERLAYS ARE LOADED'/
     45X,'LU1 -- SCRATCH FILE'/
     55X,'LU2 -- STORAGE FILE FOR STORING REVERBERATION TIME DATA'/
     65X,'LU3 -- HIGH SPEED PRINTER'/
     75X,'LU4 -- STORAGE FILE CONTAINING TABLE OF T-DISTRIBUTION'/
     85X,'LU5 -- CRT TERMINAL'//)
      9'ENTER THE DESIRED CONFIDENCE INTERVAL TO BE ATTAINED
      1 IN TERMS OF'/'PERCENT OF THE MEAN REV. TIME      (X.X)')
      READ (5,40) CINTER
      WRITE (5,30)
30    FORMAT ('ENTER THE DESIRED CONFIDENCE LEVEL IN PERCENT
     1     (XX.XX)')
      READ (5,40) CLEVEL
40    FORMAT (F10.5)
      WRITE (5,42)
42    FORMAT ('ENTER THE TIME BETWEEN SAMPLES IN SECONDS  (X.XXXXX)')
      READ (5,40) DELTH
      WRITE (5,45)
45    FORMAT ('ENTER LOWEST BAND OF INTEREST    (MIN=14)')
      READ (5,50) BLOW
50    FORMAT (I2)
      WRITE (5,55)
55    FORMAT ('ENTER HIGHEST BAND OF INTEREST    (MAX=43)')
      READ (5,50) BHIGH
      REWIND 0
      CALL IFETCH('SIGNAL',0,ISTAT)
      IF (ISTAT.NE.0) GO TO 200
      CALL SIGNAL (X,S0,S0,S1,FREQ,LINE,BLOW,BHIGH)
      GO TO 94
91    WRITE (5,92)
92    FORMAT ('YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF 200 SAMPLES')
94    WRITE (5,95)
95    FORMAT ('ENTER NO. OF SAMPLES REQUIRED FOR DECAY MEASUREMENT',
     14X,'(3 DIGITS)')
      READ (5,98)NDEC
98    FORMAT (I3)
      WRITE (5,100)
100   FORMAT ('ENTER NO. OF SAMPLES TO BE TAKEN BEFORE SOUND
     1 CUT-OFF     (3 DIGITS)')
      READ (5,98)NSIG
      N=NSIG+NDEC
      IF (N.GT.200) GO TO 91
```

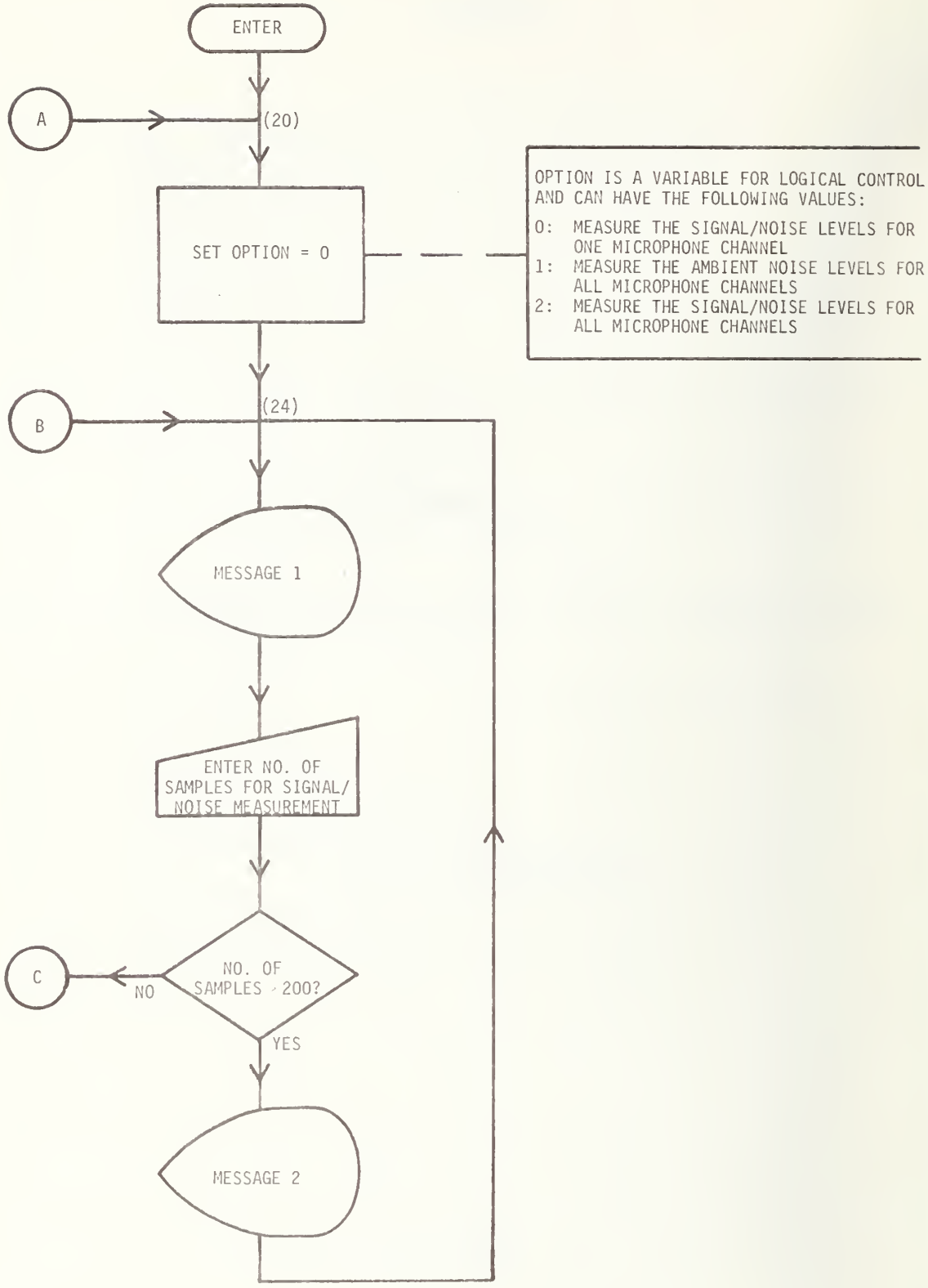
```
 61        WRITE (5,105)
 62 105    FORMAT ('ENTER UPPER CURVE FITTING LIMIT IN DB DOWN FROM
 63       1 SIGNAL LEVEL'/'(NORMALLY 5.0)')
 64        READ (5,40) CUPPER
 65        WRITE (5,106)
 66 106    FORMAT ('ENTER LOWER CURVE FITTING LIMIT IN DB UP FROM
 67       1 NOISE LEVEL'/'(NORMALLY 10.0)')
 68        READ (5,40) CLOWER
 69        REWIND 1
 70        NDCAYS=0
 71        ICRVFT=0
 72 110    AUTO=0
 73        MPLEX=0
 74 113    IF (MPLEX.EQ.1) CALL MUX(I)
 75 114    CALL SOUND(1)
 76        WRITE (5,115)
 77 115    FORMAT ('THE SOUND HAS BEEN TURNED ON'/
 78       1'ENTER "GO" TO START DECAY MEASUREMENT')
 79        READ (5,120)GS
 80 120    FORMAT (A4)
 81        GO TO 124
 82 123    INDEX=INDEX+1
 83        IF (INDEX.EQ.NAUTO) AUTO=2
 84        CALL DELAY(NDELAY)
 85 124    CALL RDMUX(MXRDG)
 86        CALL SAMPLE(NSIG,N,10,OVFLOW,X0,6060)
 87        IF (OVFLOW.EQ.1) GO TO 114
 88        IF ((MPLEX.EQ.1).AND.(AUTO.EQ.1)) CALL MUX(I)
 89        IF (AUTO.EQ.1) CALL SOUND(1)
 90        NDCAYS=NDCAYS+1
 91        WRITE (5,125) NDCAYS
 92 125    FORMAT ('DECAY NO. ',I4)
 93        IF (ICRVFT.EQ.1) GO TO 126
 94        REWIND 0
 95        CALL IFETCH('CURVFT',0,ISTAT)
 96        IF (ISTAT.NE.0) GO TO 200
 97        ICRVFT=1
 98 126    CALL CURVFT(N,NSIG,X,X0,S0,S1,ARRAY1,ARRAY2,DELTA,FREQ,
 99       1NDCAYS,AUTO,LU,BLOW,BHIGH,CUPPER,CLOWER,MM1,MM2,MXRDG)
100        IF (AUTO.EQ.1) GO TO 123
101 127    WRITE (5,128)
102 128    FORMAT (X/'DO YOU WANT TO CONDUCT ANOTHER DECAY?
103       1  (ENTER "AUTO" FOR AUTOMATIC MODE)')
104        READ (5,120)GS
105        IF (GS.EQ.'NO')GO TO 140
106        IF (GS.NE.'AUTO') GO TO 110
107 129    WRITE (5,130)
108 130    FORMAT ('ENTER NUMBER OF DECAYS TO BE CONDUCTED
109       1 AUTOMATICALLY'/'(4 DIGITS)')
110        READ (5,131) NAUTO
111 131    FORMAT (I4)
112        IF (NAUTO.LT.2) GO TO 110
113        WRITE (5,133)
114 133    FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE
115       1 FOR CURVE-FITTING PARAMETERS')
116        READ (5,135) LU
117 135    FORMAT (I1)
118        WRITE (5,136)
119 136    FORMAT ('ENTER NO. OF SECONDS OF DELAY REQUIRED FOR SOUND
120       1 BUILD-UP    (XX.X)'/'(MAX = 32)'/'(THIS DELAY IS IN
```

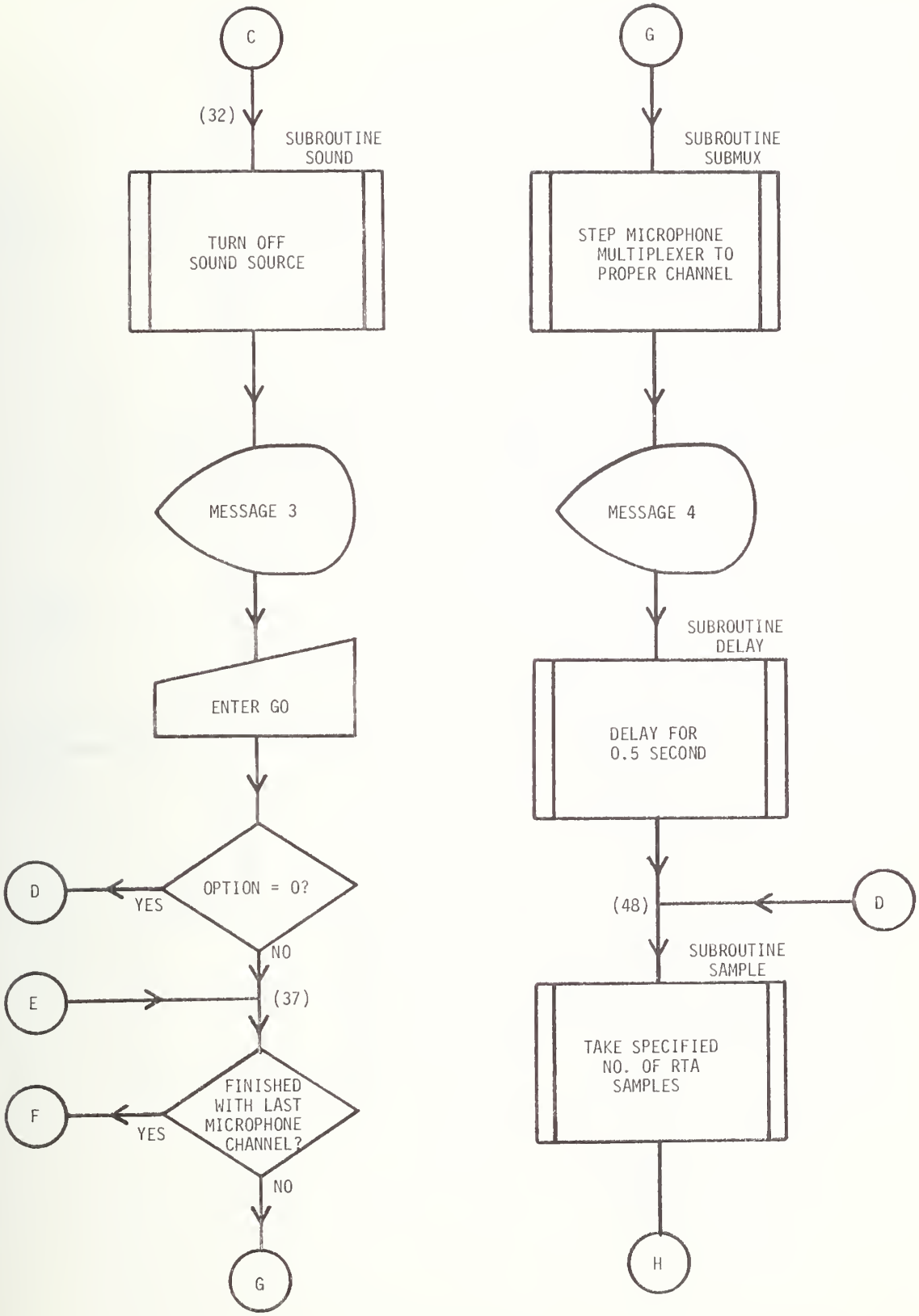
```
      2 ADDITION TO THE COMPUTING AND PRINTOUT TIME)')
       READ (5,40) TDELAY
       NDELAY=TDELAY*1000.+.5
       WRITE (5,138)
138    FORMAT ('DO YOU WANT TO STEP THE MICROPHONE MULTIPLEXER
      1 BEFORE EACH DECAY?')
       READ (5,120) GS
       MPLEX=1
       IF (GS.EQ.'NO') MPLEX=0
       AUTO=1.
       INDEX=1
       GO TO 113
140    ICRVFT=0
       WRITE (5,150)
150    FORMAT ('DO YOU WANT TO SEE THE DATA POINTS OF THE LAST DECAY?')
       READ (5,120)GS
       IF (GS.EQ.'NO')GO TO 160
       REWIND 0
       CALL IFETCH('DSDATA',0,ISTAT)
       IF (ISTAT.NE.0) GO TO 200
       CALL DSDATA(X,X0,ARRAY1,ARRAY2,FREQ,N,NDCAYS,LINE,DELTA,
      1S0,S1,CUPPER,CLOWER,MM1,MM2,MXRDG)
160    WRITE (5,170)
170    FORMAT ('DO YOU WANT TO SEE THE AVERAGES?')
       READ (5,120)GS
       IF (GS.EQ.'NO')GO TO 127
       REWIND 0
       CALL IFETCH('AVRGE',0,ISTAT)
       IF (ISTAT.NE.0)  GO TO 200
       CALL AVRGE(X,X0,ARRAY1,ARRAY2,NDCAYS,CINTER,CLEVEL,FREQ,
      1BLOW,BHIGH)
180    WRITE (5,128)
       READ (5,120)GS
       IF (GS.EQ.'AUTO') GO TO 129
       IF (GS.NE.'NO')GO TO 110
       REWIND 0
       CALL IFETCH('RSULTS',0,ISTAT)
       IF (ISTAT.NE.0) GO TO 200
       CALL RSULTS(X,X0,ARRAY1,ARRAY2,FREQ,NDCAYS,CLEVEL,NSIG,
      1NDEC,LINE,DELTA,BLOW,BHIGH,CUPPER,CLOWER)
       GO TO 220
200    WRITE (5,210) ISTAT
210    FORMAT ('OVERLAY LOAD ERROR ',I4,' -- TASK ABORTED')
220    WRITE (5,230)
230    FORMAT (' END OF JOB')
       END
```

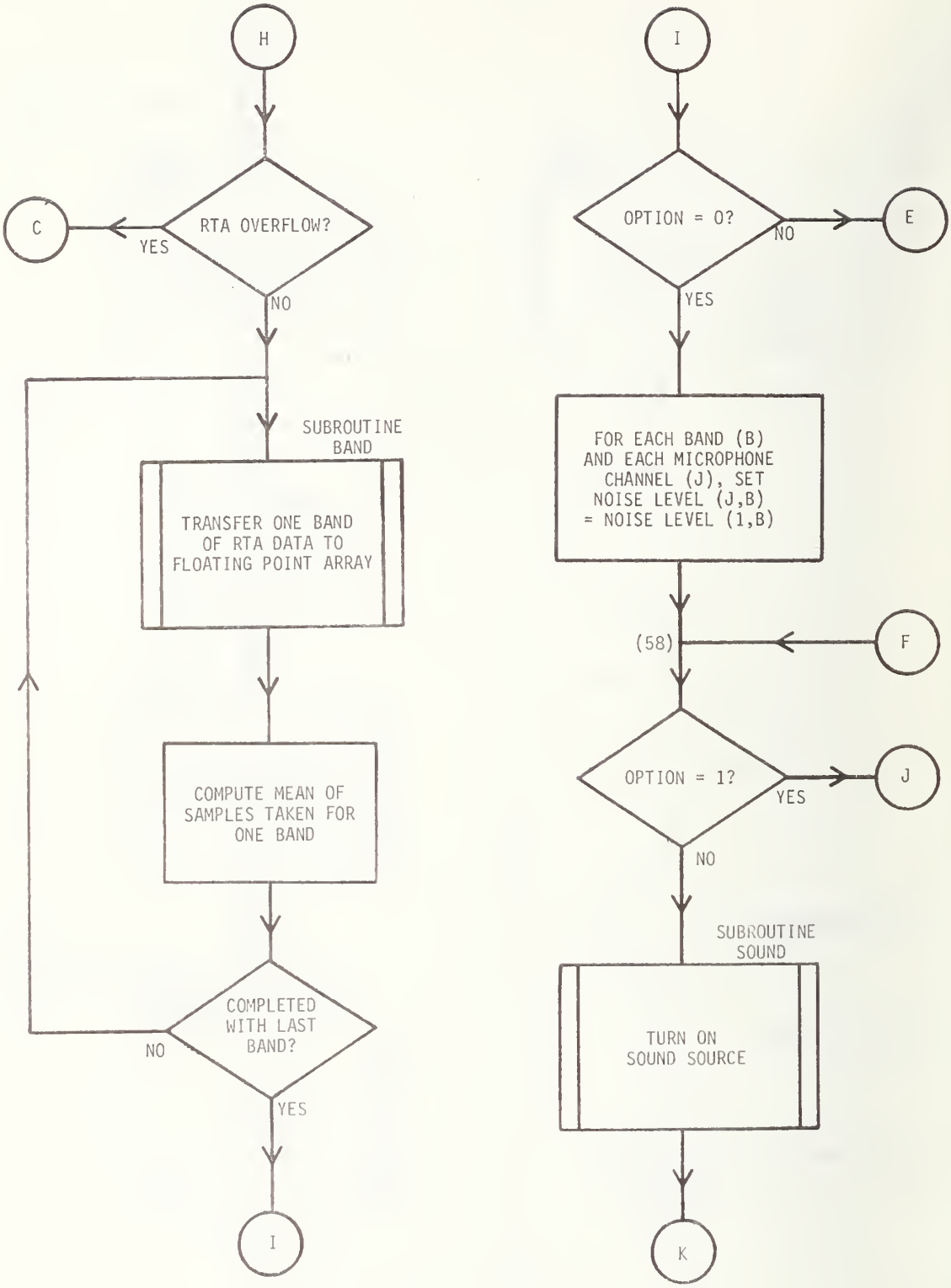
APPENDIX C

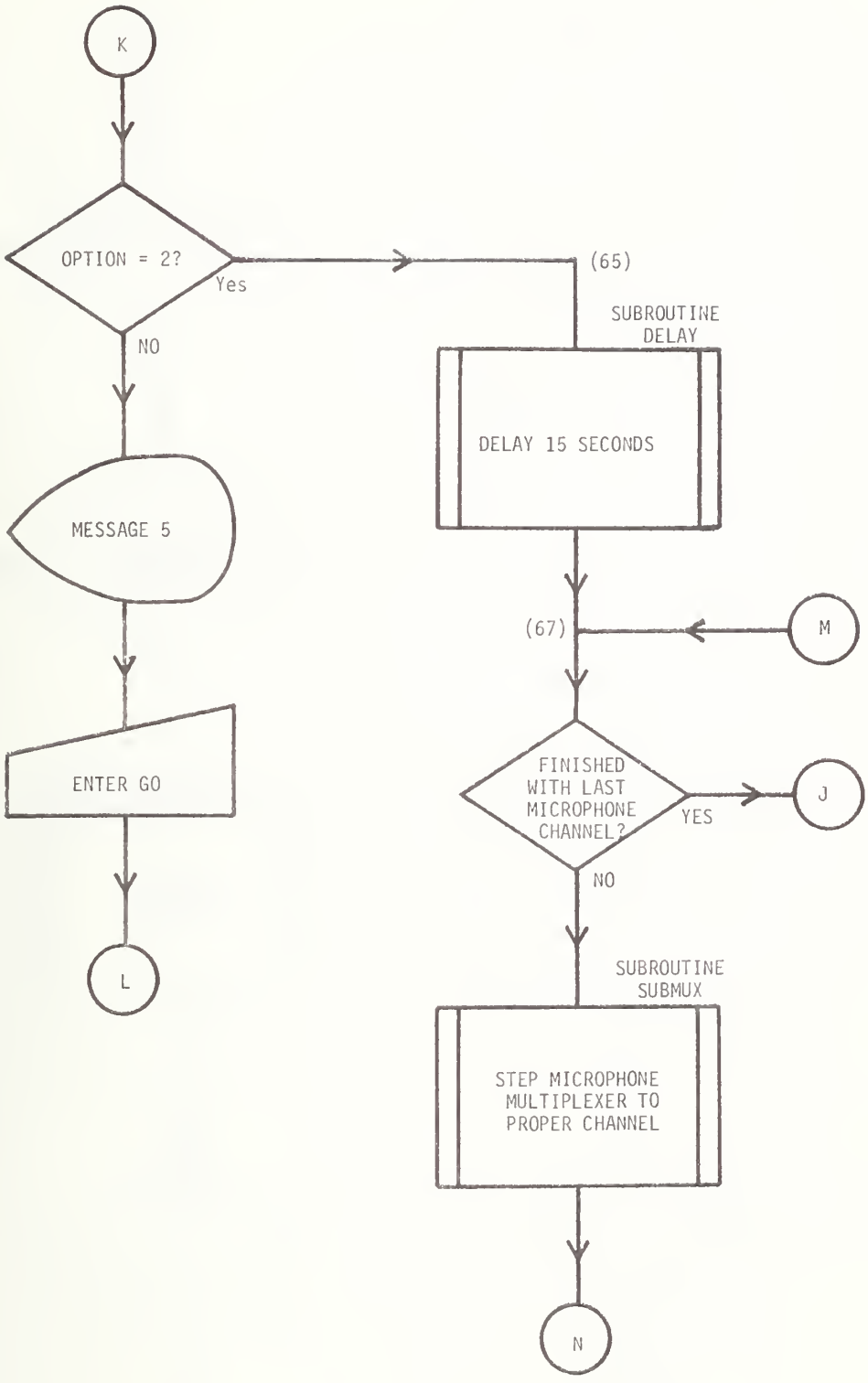
Subroutine SIGNAL Flow Chart, Terminal Messages, Printouts, and Listings

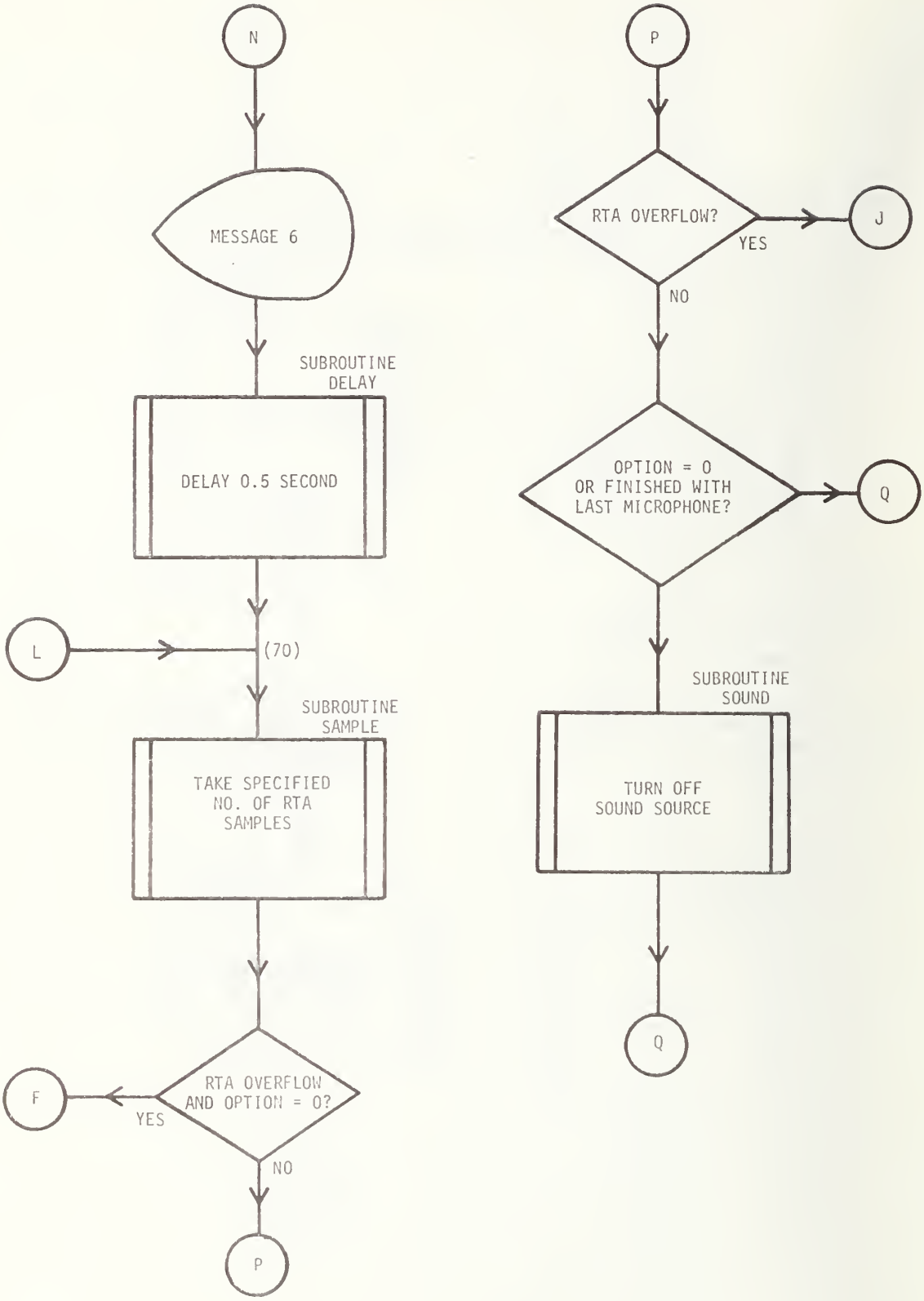
FLOW CHART: SUBROUTINE SIGNAL

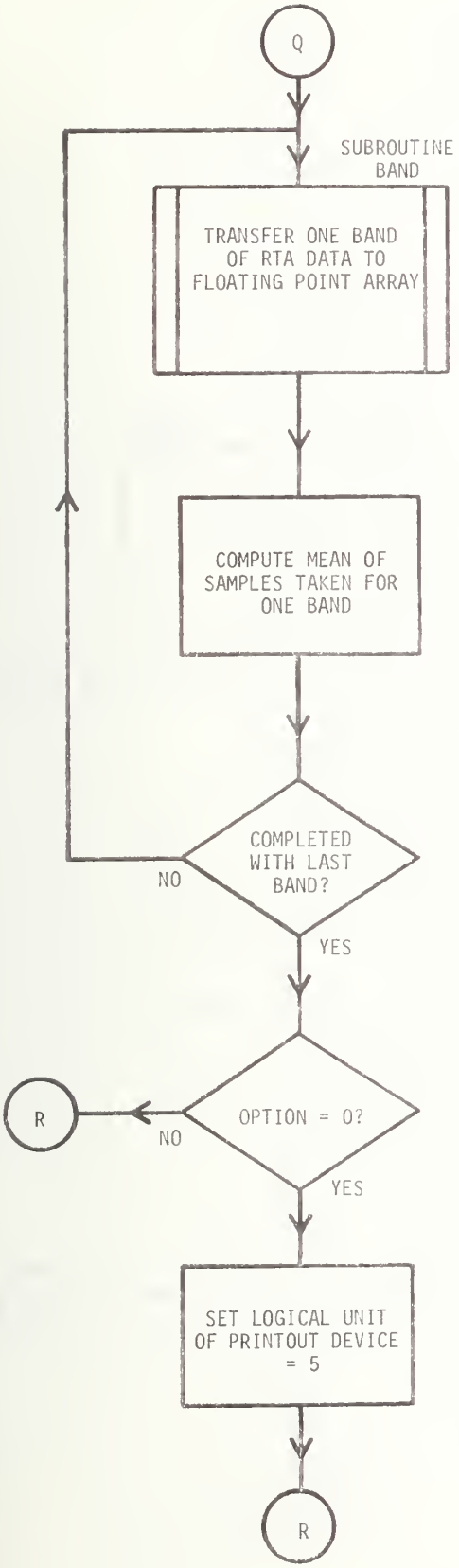


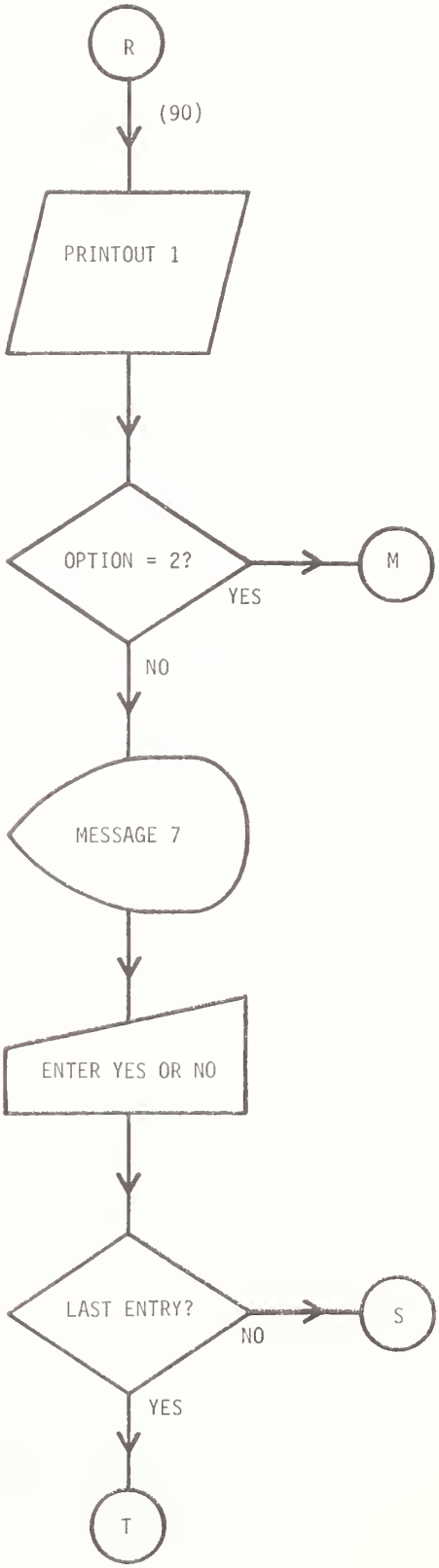

C
(32)
SUBROUTINE
SOUND
TURN OFF
SOUND SOURCE
MESSAGE 3
ENTER GO
OPTION = 0?
D
YES
NO
E
(37)
FINISHED
WITH LAST
MICROPHONE
CHANNEL?
F
YES
NO
G
G
SUBROUTINE
SUBMUX
STEP MICROPHONE
MULTIPLEXER TO
PROPER CHANNEL
MESSAGE 4
SUBROUTINE
DELAY
DELAY FOR
0.5 SECOND
(48)
D
SUBROUTINE
SAMPLE
TAKE SPECIFIED
NO. OF RTA
SAMPLES
H


H
RTA OVERFLOW?
C
YES
NO
SUBROUTINE
BAND
TRANSFER ONE BAND
OF RTA DATA TO
FLOATING POINT ARRAY
COMPUTE MEAN OF
SAMPLES TAKEN FOR
ONE BAND
COMPLETED
WITH LAST
BAND?
NO
YES
I
I
OPTION = 0?
NO
E
YES
FOR EACH BAND (B)
AND EACH MICROPHONE
CHANNEL (J), SET
NOISE LEVEL (J,B)
= NOISE LEVEL (1,B)
(58)
F
OPTION = 1?
YES
J
NO
SUBROUTINE
SOUND
TURN ON
SOUND SOURCE
K

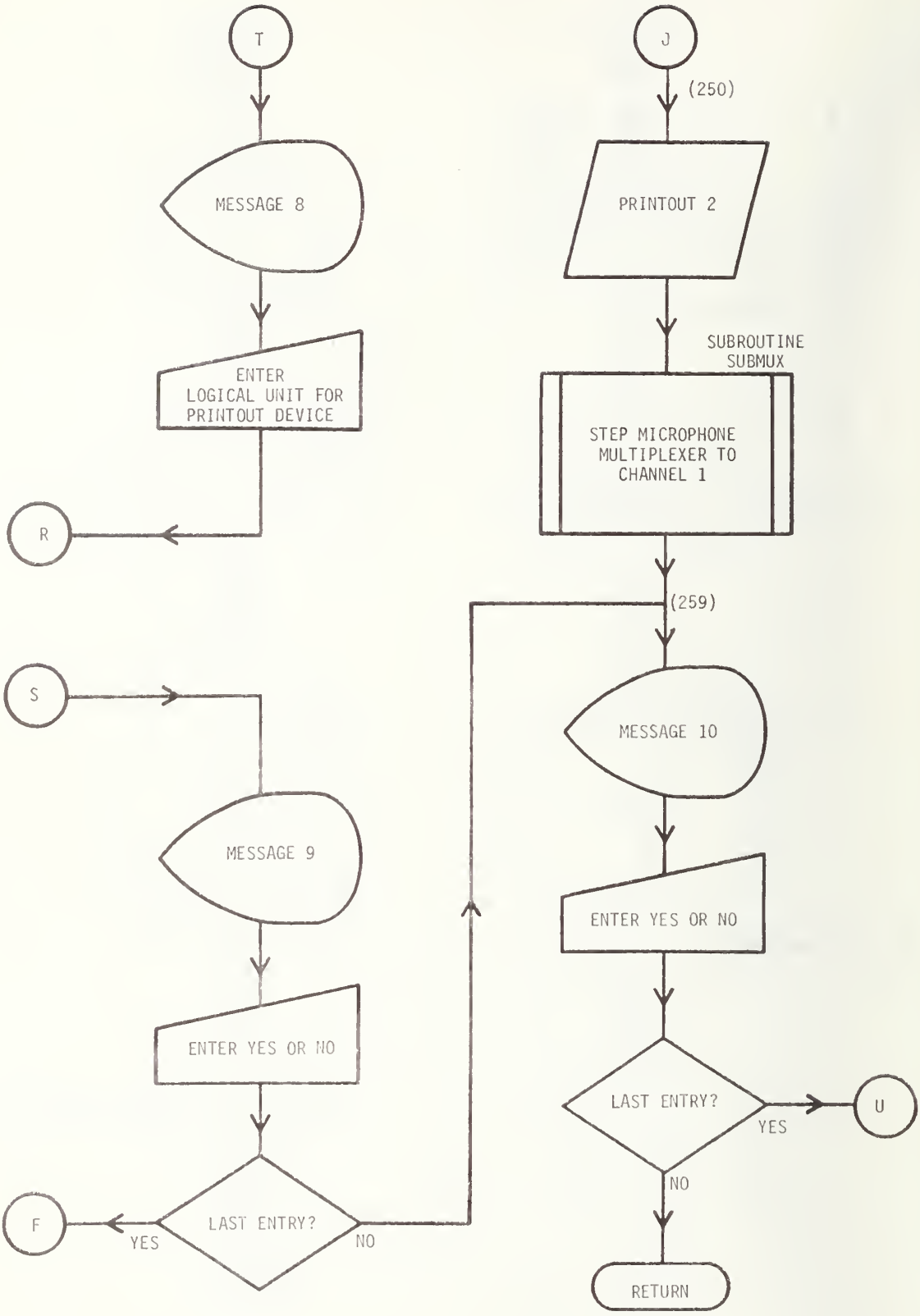

K
OPTION = 2?
Yes
NO
(65)
SUBROUTINE
DELAY
DELAY 15 SECONDS
MESSAGE 5
(67)
M
ENTER GO
FINISHED
WITH LAST
MICROPHONE
CHANNEL?
YES
J
NO
L
SUBROUTINE
SUBMUX
STEP MICROPHONE
MULTIPLEXER TO
PROPER CHANNEL
N

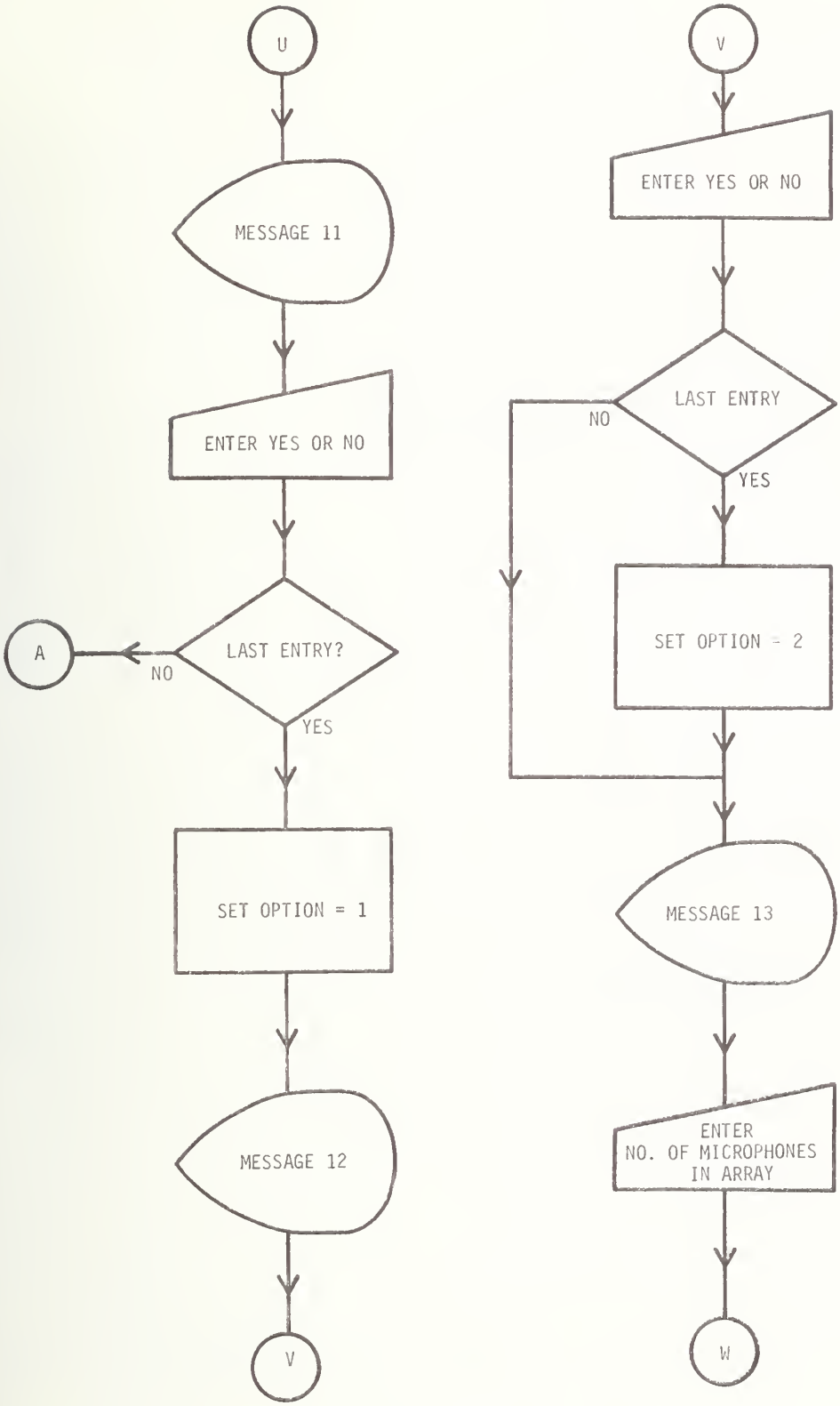

N
MESSAGE 6
SUBROUTINE DELAY
DELAY 0.5 SECOND
L
(70)
SUBROUTINE SAMPLE
TAKE SPECIFIED NO. OF RTA SAMPLES
F
RTA OVERFLOW AND OPTION = 0?
YES
NO
P
P
RTA OVERFLOW?
YES
J
NO
OPTION = 0 OR FINISHED WITH LAST MICROPHONE?
Q
SUBROUTINE SOUND
TURN OFF SOUND SOURCE
Q


Q
SUBROUTINE BAND
TRANSFER ONE BAND OF RTA DATA TO FLOATING POINT ARRAY
COMPUTE MEAN OF SAMPLES TAKEN FOR ONE BAND
COMPLETED WITH LAST BAND?
NO
YES
OPTION = 0?
R
NO
YES
SET LOGICAL UNIT OF PRINTOUT DEVICE = 5
R

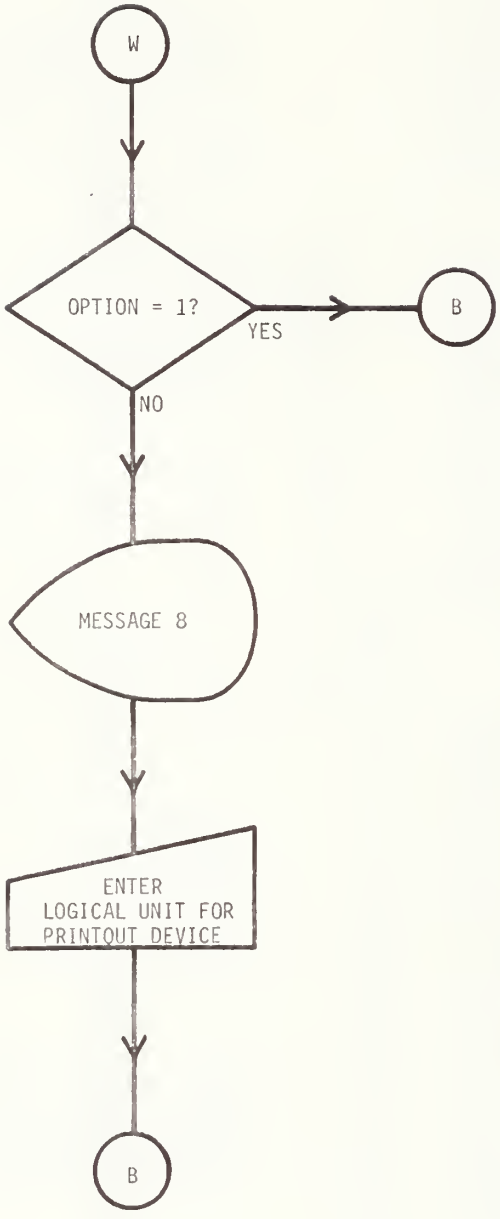

R
(90)
PRINTOUT 1
OPTION = 2?
M
YES
NO
MESSAGE 7
ENTER YES OR NO
LAST ENTRY?
S
NO
YES
T

SIGNAL - 7


T
MESSAGE 8
ENTER
LOGICAL UNIT FOR
PRINTOUT DEVICE
R
J
(250)
PRINTOUT 2
SUBROUTINE
SUBMUX
STEP MICROPHONE
MULTIPLEXER TO
CHANNEL 1
(259)
MESSAGE 10
ENTER YES OR NO
LAST ENTRY?
YES
U
NO
RETURN
S
MESSAGE 9
ENTER YES OR NO
F
LAST ENTRY?
YES
NO


U
MESSAGE 11
ENTER YES OR NO
A
LAST ENTRY?
NO
YES
SET OPTION = 1
MESSAGE 12
V
V
ENTER YES OR NO
LAST ENTRY
NO
YES
SET OPTION = 2
MESSAGE 13
ENTER
NO. OF MICROPHONES
IN ARRAY
W

SIGNAL - 9

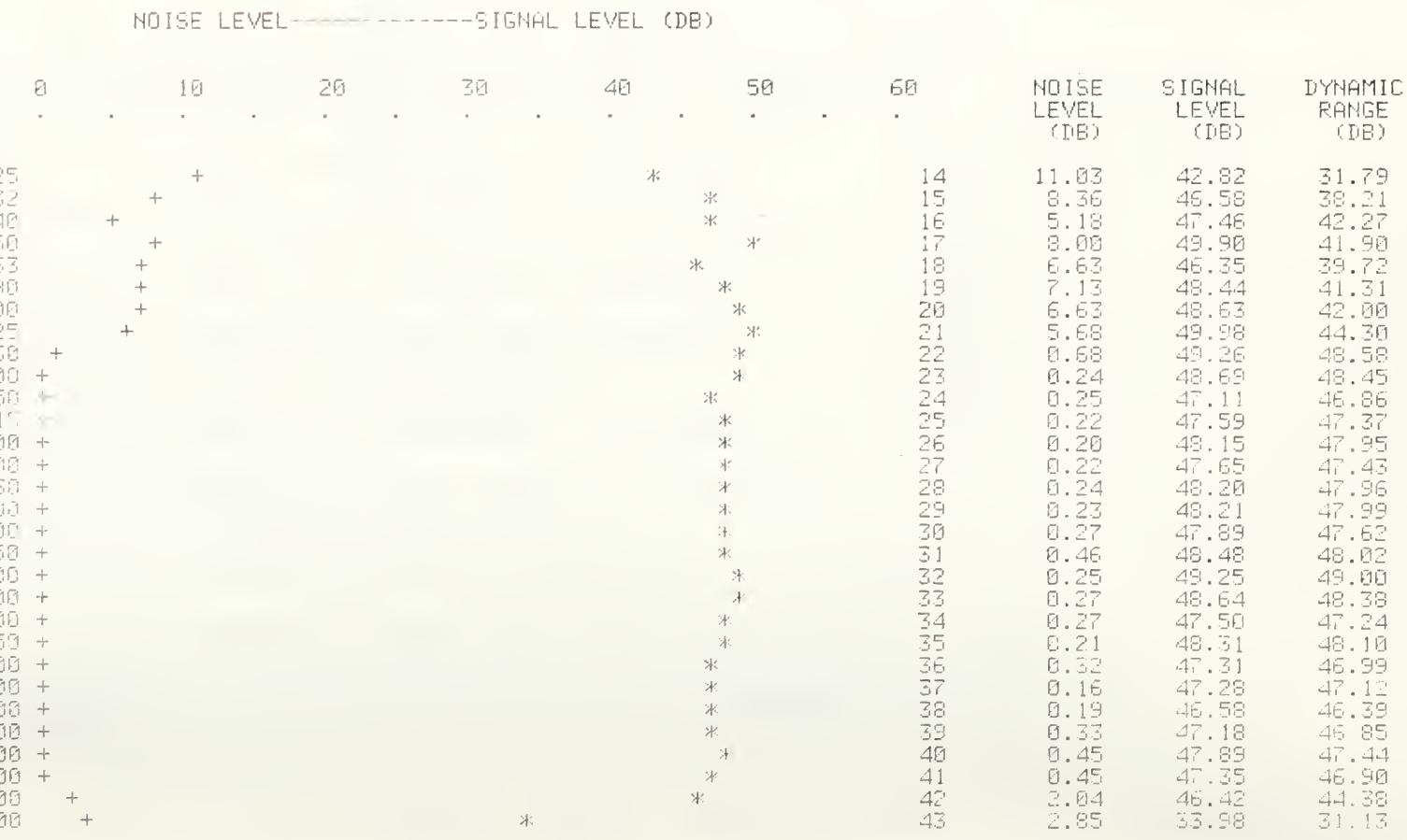

W
OPTION = 1?
YES
B
NO
MESSAGE 8
ENTER
LOGICAL UNIT FOR
PRINTOUT DEVICE
B

CRT TERMINAL MESSAGES: SUBROUTINE SIGNAL

MESSAGE NO.	FORMAT NO.	MESSAGE
1	25	ENTER NO. OF SAMPLES FOR SIGNAL/NOISE MEASUREMENT (3 DIGITS)
2	28	YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF 200 SAMPLES
3	35	THE SOUND HAS BEEN TURNED OFF ENTER "GO" TO START AMBIENT NOISE MEASUREMENT
4	38	AMBIENT NOISE MEASUREMENT, MICROPHONE 1
5	60	THE SOUND HAS BEEN TURNED ON ENTER "GO" TO START SIGNAL MEASUREMENT
6	68	SIGNAL LEVEL MEASUREMENT, MICROPHONE 1
7	200	DO YOU WANT ANOTHER PRINT OF THESE RESULTS?
8	210	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE
9	240	DO YOU WANT TO REPEAT THE SIGNAL MEASUREMENT?
10	260	DO YOU WANT TO REPEAT THE SIGNAL AND NOISE MEASUREMENT?
11	270	DO YOU WANT A SEPARATE AMBIENT NOISE MEASUREMENT FOR EACH MICROPHONE CHANNEL?
12	275	DO YOU WANT TO INCLUDE A SIGNAL LEVEL MEASUREMENT FOR EACH CHANNEL?
13	280	ENTER NO. OF MICROPHONES IN THE MICROPHONE ARRAY (2 DIGITS)

PRINTOUT 1 - SUBROUTINE SIGNAL

SIGNAL/NOISE MEASUREMENT: NO. OF SAMPLES = 200



BAND NO.	NOISE LEVEL (DB)	SIGNAL LEVEL (DB)	DYNAMIC RANGE (DB)
14	11.03	42.82	31.79
15	8.36	46.58	38.21
16	5.18	47.46	42.27
17	8.00	49.90	41.90
18	6.63	46.35	39.72
19	7.13	48.44	41.31
20	6.63	48.63	42.00
21	5.68	49.98	44.30
22	0.68	49.26	48.58
23	0.24	48.69	48.45
24	0.25	47.11	46.86
25	0.22	47.59	47.37
26	0.20	48.15	47.95
27	0.22	47.65	47.43
28	0.24	48.20	47.96
29	0.23	48.21	47.99
30	0.27	47.89	47.62
31	0.46	48.48	48.02
32	0.25	49.25	49.00
33	0.27	48.64	48.38
34	0.27	47.50	47.24
35	0.21	48.31	48.10
36	0.32	47.31	46.99
37	0.16	47.28	47.12
38	0.19	46.58	46.39
39	0.33	47.18	46.85
40	0.45	47.89	47.44
41	0.45	47.35	46.90
42	3.04	46.42	44.38
43	2.85	33.98	31.13

PRINTOUT 2 - SUBROUTINE SIGNAL

AMBIENT NOISE LEVELS

BAND NO.	FREQ.	MIC 1	MIC 2	MIC 3	MIC 4	MIC 5	MIC 6	MIC 7	MIC 8	MIC 9	MIC 10	MIC 11	MIC 12
14	25	10.11	15.04	8.81	3.74	13.49	4.44	19.76	9.36	23.85	14.95	19.54	20.69
15	32	8.29	12.42	6.99	3.76	13.15	8.38	19.16	10.82	22.69	12.50	17.63	21.97
16	40	5.87	13.76	4.98	1.40	12.05	0.21	17.77	6.80	20.75	11.78	18.30	20.08
17	50	8.88	15.82	8.78	7.11	11.85	2.10	15.99	7.32	19.05	12.54	15.19	18.09
18	63	6.57	11.21	3.76	3.95	9.41	3.66	16.26	6.89	16.43	8.55	12.70	15.84
19	80	8.31	11.83	4.36	6.48	9.43	5.71	12.34	5.24	13.06	7.17	10.55	12.74
20	100	7.80	7.99	6.96	7.15	7.06	5.14	9.98	5.31	9.65	6.52	8.89	10.17
21	125	6.01	8.57	7.07	7.46	8.25	4.00	7.47	4.65	5.68	5.44	8.93	6.97
22	160	0.61	1.38	0.07	0.64	1.13	0.11	1.96	0.18	1.49	0.31	0.95	1.59
23	200	0.09	0.25	0.00	0.05	0.04	0.04	0.56	0.01	0.18	0.02	0.27	0.16
24	250	0.06	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.00	0.05	0.00
25	315	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
26	400	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	500	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	630	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00
29	800	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	1000	0.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	1250	0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
32	1600	0.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
33	2000	0.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
34	2500	0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
35	3150	0.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00
36	4000	0.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
37	5000	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00
38	6300	0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
39	8000	0.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00
40	10000	0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00
41	12500	0.51	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.19	0.00	0.00	0.10
42	16000	2.28	1.55	1.44	1.22	1.85	1.17	1.16	1.11	2.36	1.39	1.47	1.86
43	20000	2.51	2.51	2.55	3.00	2.65	2.79	0.00	1.92	3.34	2.60	1.97	0.00

NO. OF SAMPLES = 200
NO. OF MICROPHONES = 12

SUBROUTINE SIGNAL

```
$ASSM
SIGNAL PROG SIGNAL - SUBROUTINE TO DETERMINE SIGNAL/NOISE LEVELS
$FORT
C    ------------------------------
C    OTHER SUBROUTINES CALLED:  SAMPLE,BAND,SOUND,SUBMUX,MUX,DELAY
C    ------------------------------
      SUBROUTINE SIGNAL(X,X0,S0,S1,FREQ,LINE,BLOW,BHIGH)
      INTEGER*2 B,N,X0(6060),FREQ(30),LU,BELL,FF,OPTION,
     1BLOW,BHIGH,B1,B2,OVFLOW,LINE(61),BLANK,STAR,PLUS
      DIMENSION X(200),S0(30,12),S1(30)
      DATA BLANK,STAR,PLUS,FF/X'2000',X'2A00',X'2B00',
     1X'0C00'/
20    OPTION=0
24    WRITE (5,25)
25    FORMAT ('ENTER NO. OF SAMPLES FOR SIGNAL/NOISE MEASUREMENT'
     12X,'(3 DIGITS)')
      READ (5,26)N
26    FORMAT (I3)
      IF (N.LE.200)GO TO 32
      WRITE (5,28)
28    FORMAT ('YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF 200 SAMPLES')
      GO TO 24
32    CALL SOUND(0)
34    WRITE (5,35)
35    FORMAT ('THE SOUND HAS BEEN TURNED OFF'/
     1'ENTER "GO" TO START AMBIENT NOISE MEASUREMENT')
      READ (5,36)GS
36    FORMAT (A4)
      J=1
      IF (OPTION.EQ.0) GO TO 48
      J=0
37    J=J+1
      IF (J.GT.NMIC) GO TO 58
      CALL SUBMUX(J)
      WRITE (5,38) J
38    FORMAT ('AMBIENT NOISE MEASUREMENT.  MICROPHONE ',I2)
      CALL DELAY(500)
48    CALL SAMPLE(0,N,10,OVFLOW,X0,6060)
      IF (OVFLOW.EQ.1) GO TO 32
      DO 50 B=BLOW,BHIGH
      IB=B-13
      CALL BAND(B,N,X,X0,6060)
      C=0.
      DO 49 I=1,N
      C=C+10.**(X(I)/10.)
49    CONTINUE
      C=C/N
      S0(IB,J)=10.*ALOG10(C)
50    CONTINUE
      IF (OPTION.NE.0) GO TO 37
      DO 55 J=2,12
      DO 55 B=BLOW,BHIGH
      IB=B-13
      S0(IB,J)=S0(IB,1)
55    CONTINUE
58    IF (OPTION.EQ.1) GO TO 250
      CALL SOUND(1)
      IF (OPTION.EQ.2) GO TO 65
      WRITE (5,60)
60    FORMAT ('THE SOUND HAS BEEN TURNED ON'
```

```
      1'ENTER "GO" TO START SIGNAL MEASUREMENT')
       READ (5,36)GS
       GO TO 70
65     CALL DELAY(15000)
       J=0
67     J=J+1
       IF (J.GT.NMIC) GO TO 250
       CALL SUBMUX(J)
       WRITE (5,68) J
68     FORMAT ('SIGNAL  LEVEL MEASUREMENT,  MICROPHONE ',I2)
       CALL DELAY(500)
70     CALL SAMPLE(N,X,10,OVFLOW,X0,6060)
       IF ((OVFLOW.EQ.1).AND.(OPTION.EQ.0)) GO TO 58
       IF (OVFLOW.EQ.1) GO TO 250
       IF ((OPTION.EQ.0).OR.(J.EQ.NMIC)) CALL SOUND(0)
       DO 80 B=BLOW,BHIGH
       IB=B-13
       CALL BAND(B,N,X,X0,6060)
       C=0.
       DO 75 I=1,N
       C=C+10.**(X(I)/10.)
75     CONTINUE
       C=C/N
       S1(IB)=10.*ALOG10(C)
80     CONTINUE
       IF (OPTION.EQ.0) LU=5
90     WRITE (LU,92) N
92     FORMAT ('SIGNAL/NOISE MEASUREMENT:',
      1T30,'NO. OF SAMPLES = ',I3)
       IF (OPTION.EQ.0) GO TO 98
       WRITE (LU,94) J
94     FORMAT (T30,'MICROPHONE NO. ',I2)
98     WRITE (LU,100)
100    FORMAT (5(/),13X,'NOISE LEVEL-------------SIGNAL LEVEL (DB)')
       IF (LU.EQ.5) GO TO 105
       WRITE (LU,102)
102    FORMAT (/,/,6X,'0',9X,'10',8X,'20',8X,'30',8X,'40',
      18X,'50',8X,'60',T77,'NOISE',T86,'SIGNAL',T96,'DYNAMIC'/
      26X,13('.',4X),T77,'LEVEL',T87,'LEVEL',T97,'RANGE'/
      3T78,'(DB)',T88,'(DB)',T98,'(DB)'/)
       GO TO 108
105    WRITE (5,106)
106    FORMAT (/,/,6X,'0',9X,'10',8X,'20',8X,'30',8X,'40',
      18X,'50',8X,'60'/6X,13('.',4X)/,/)
108    DO 110 I=1,61
       LINE(I)=BLANK
110    CONTINUE
       DO 130 B=BLOW,BHIGH
       IB=B-13
       J1=S0(IB,J)+1.5
       J2=S1(IB)+1.5
       LINE(J1)=PLUS
       LINE(J2)=STAR
       IF (LU.EQ.5) GO TO 125
       C=S1(IB)-S0(IB,J)
       WRITE (LU,120)FREQ(IB),(LINE(I),I=1,61),B,S0(IB,J),S1(IB),C
120    FORMAT (I5,X,61A1,X,I2,T76,F6.2,T86,F6.2,T96,F6.2)
       GO TO 128
125    WRITE (5,126) FREQ(IB),(LINE(I),I=1,61),B
126    FORMAT (I5,X,61A1,X,I2)
```

SIGNAL-3

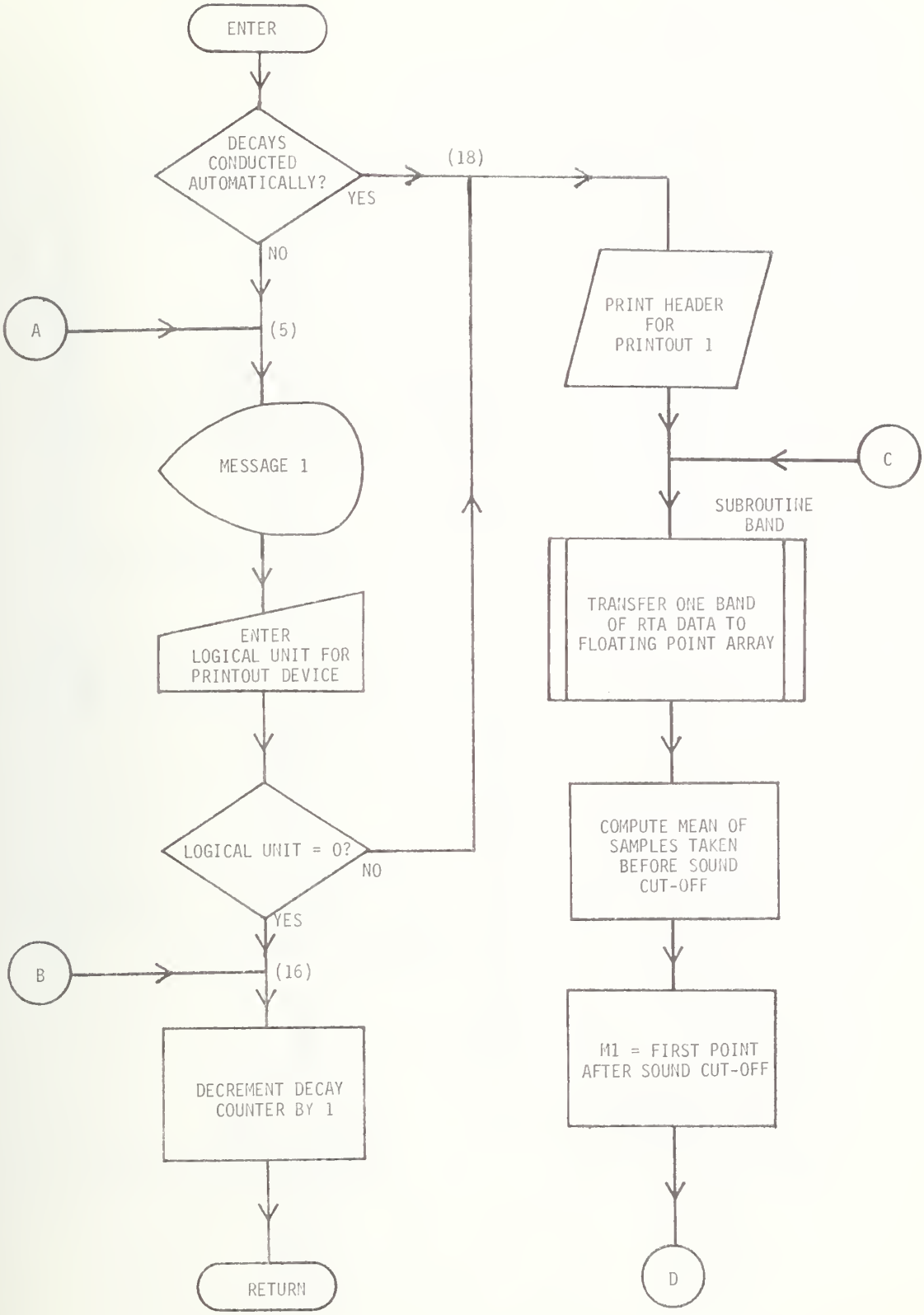
```
      LINE(J1)=BLANK
      LINE(J2)=BLANK
  130 CONTINUE
      WRITE (LU,140)FF
  140 FORMAT (X,'FREQUENCY',36X,'BAND NO.',2X,'(HZ)',A1)
      IF (OPTION.EQ.2) GO TO 67
      WRITE (5,200)
  200 FORMAT ('DO YOU WANT ANOTHER PRINT OF THESE RESULTS?')
      READ (5,36)GS
      IF (GS.EQ.'NO')GO TO 230
      WRITE (5,210)
  210 FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE')
      READ (5,220)LU
  220 FORMAT (I1)
      GO TO 30
  230 WRITE (5,240)
  240 FORMAT ('DO YOU WANT TO REPEAT THE SIGNAL MEASUREMENT?')
      READ (5,36)GS
      IF (GS.NE.'NO') GO TO 58
      GO TO 359
  250 WRITE (5,252) (I,I=1,12)
  252 FORMAT (//,T4,'AMBIENT NOISE LEVELS',//,X,X,X,
     1'BAND',T2,'NO.',T8,'FREQ.',T16,'MIC ',I1,3X),
     2 'MIC ',I1,2X,' )
      DO 257 B=BLOW,BHIGH
      IE(B-1)
      T1=0.0
      DO 254 I=1,12
      SO(B,I)=0.
  254 CONTINUE
      WRITE (5,255) B,(SO(B,I),I=1,12)
  255 FORMAT (T3,I2,T8,12(F6.2,2X))
  257 CONTINUE
      WRITE (5,258) FF,FF
  258 FORMAT (A1,//,' NO. OF SAMPLES = ',I3,
     1'NO. OF MICROPHONES = ',I2,A1)
      CALL SUBROUTINE
  259 WRITE (5,260)
  260 FORMAT (T4,'DO YOU WANT TO REPEAT THE SIGNAL AND NOISE
     1 MEASUREMENT')
      READ (5,36)GS
      IF (GS.NE.'NO') GO TO 400
      WRITE (5,270)
  270 FORMAT (' DO YOU WANT A SEPARATE AMBIENT NOISE MEASUREMENT'
     1,' FOR EACH CHANNEL')
      READ (5,36)GS
      IF (GS.NE.'NO') GO TO 30
      OPTION=2
      WRITE (5,275)
  275 FORMAT (' DO YOU WANT TO MODIFY A SIGNAL LEVEL MEASUREMENT'
     1,' FROM BEFORE')
      READ (5,36) GS
      IF (GS.NE.'NO') GO TO 67-2
      WRITE (5,280)
  280 FORMAT (' ENTER NO. OF MICROPHONES IN THE MICROPHONE ARRAY'
     1,' (1-12)')
      READ (5,290) NMIC
  290 FORMAT (I2)
      IF (NMIC.GT.12.OR.NMIC.LT.1) GO TO 30
      WRITE (5,300)
```

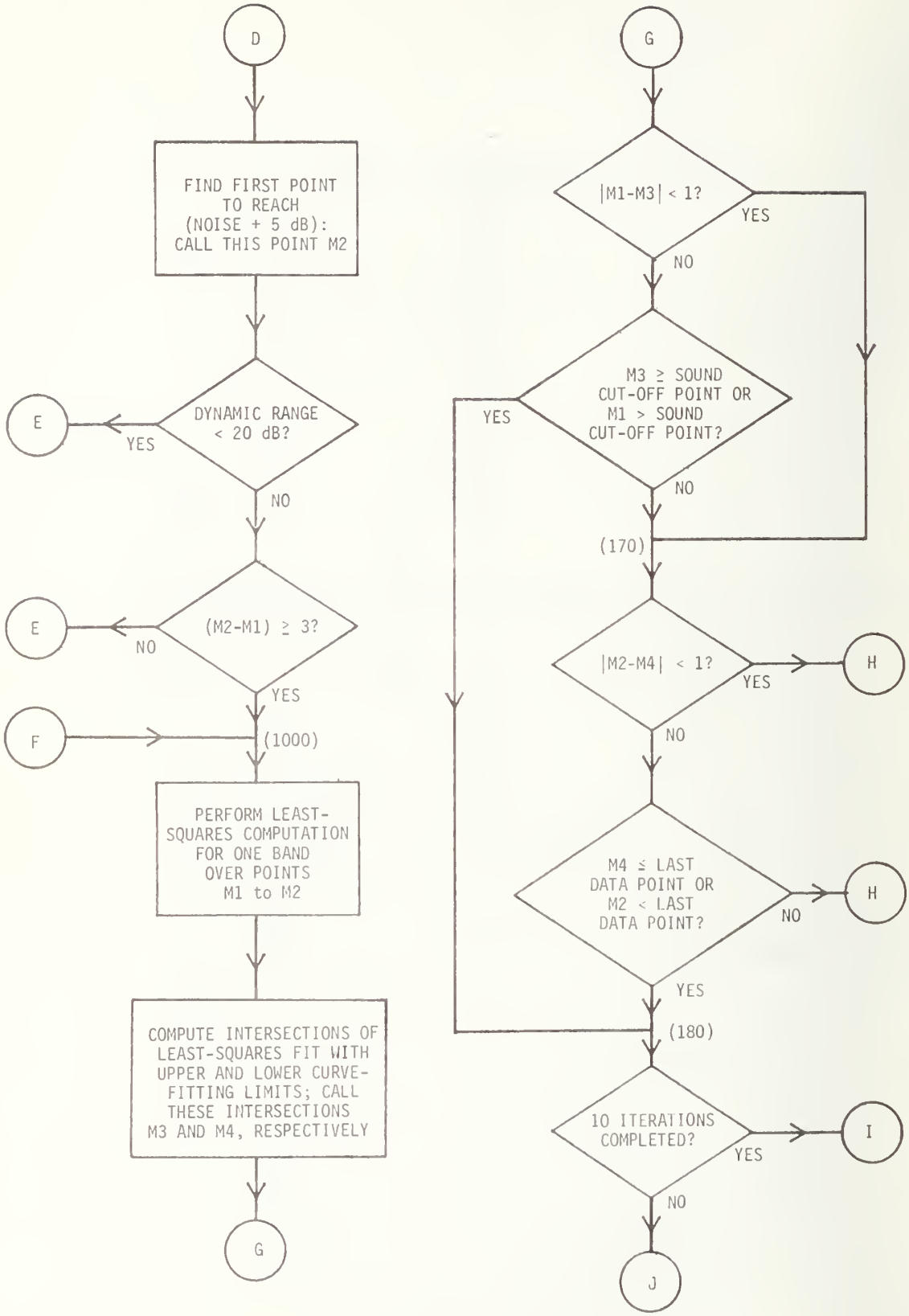
```
181         READ (5,220) LU
182         GO TO 24
183 400     RETURN
184         END
```

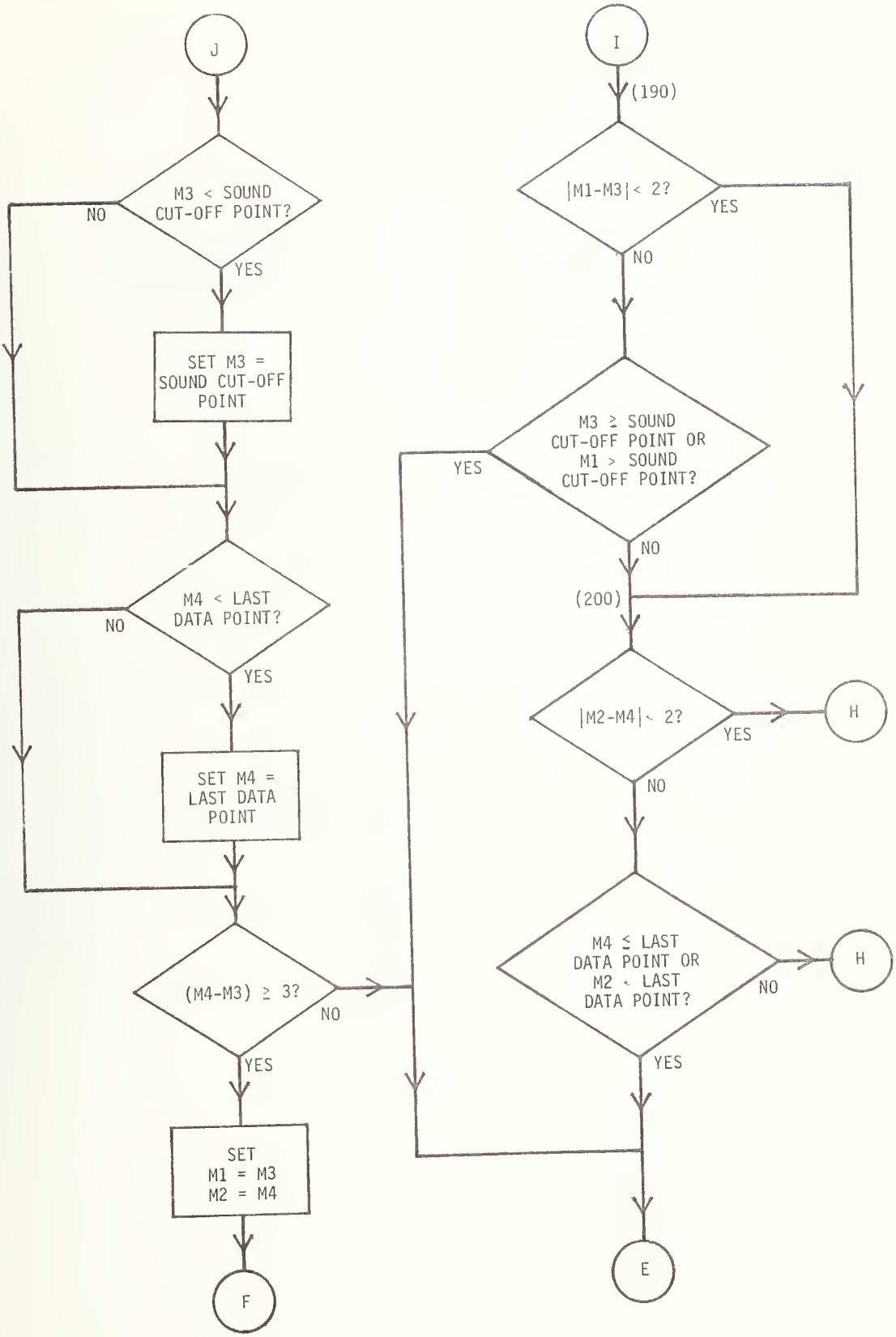
APPENDIX D

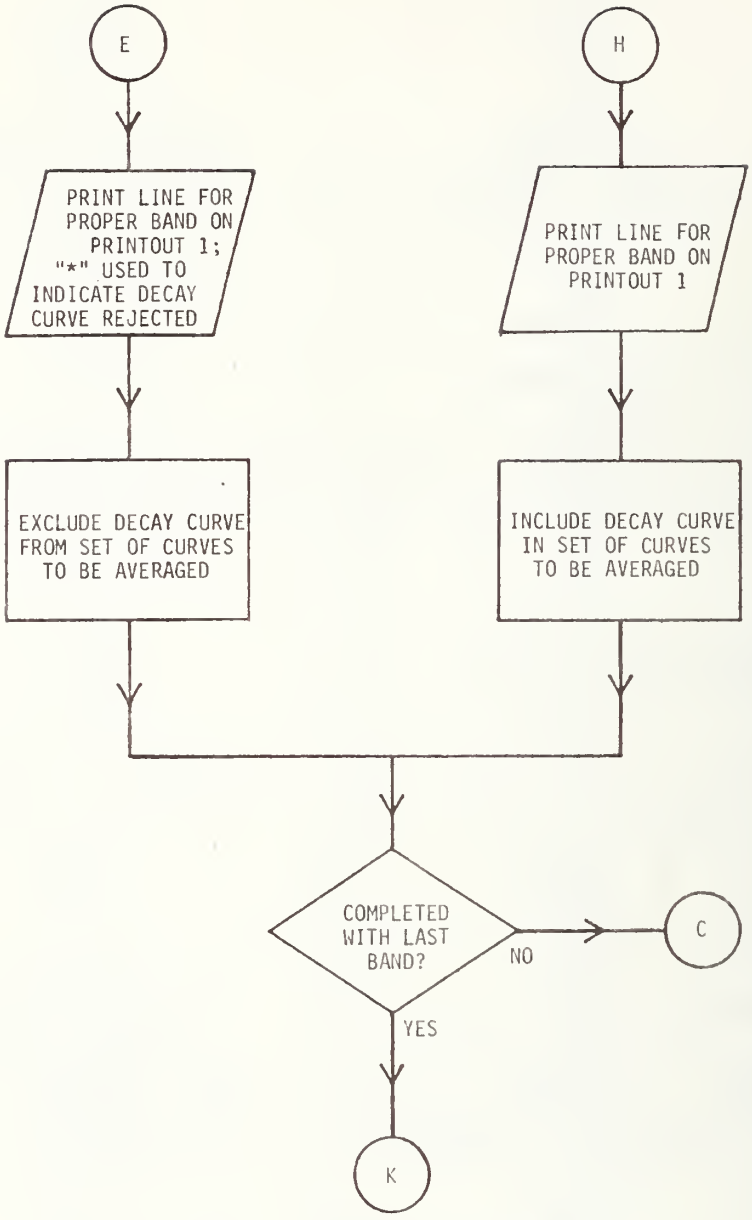
Subroutine CURVFT Flow Chart, Terminal Messages, Printout, and Listings

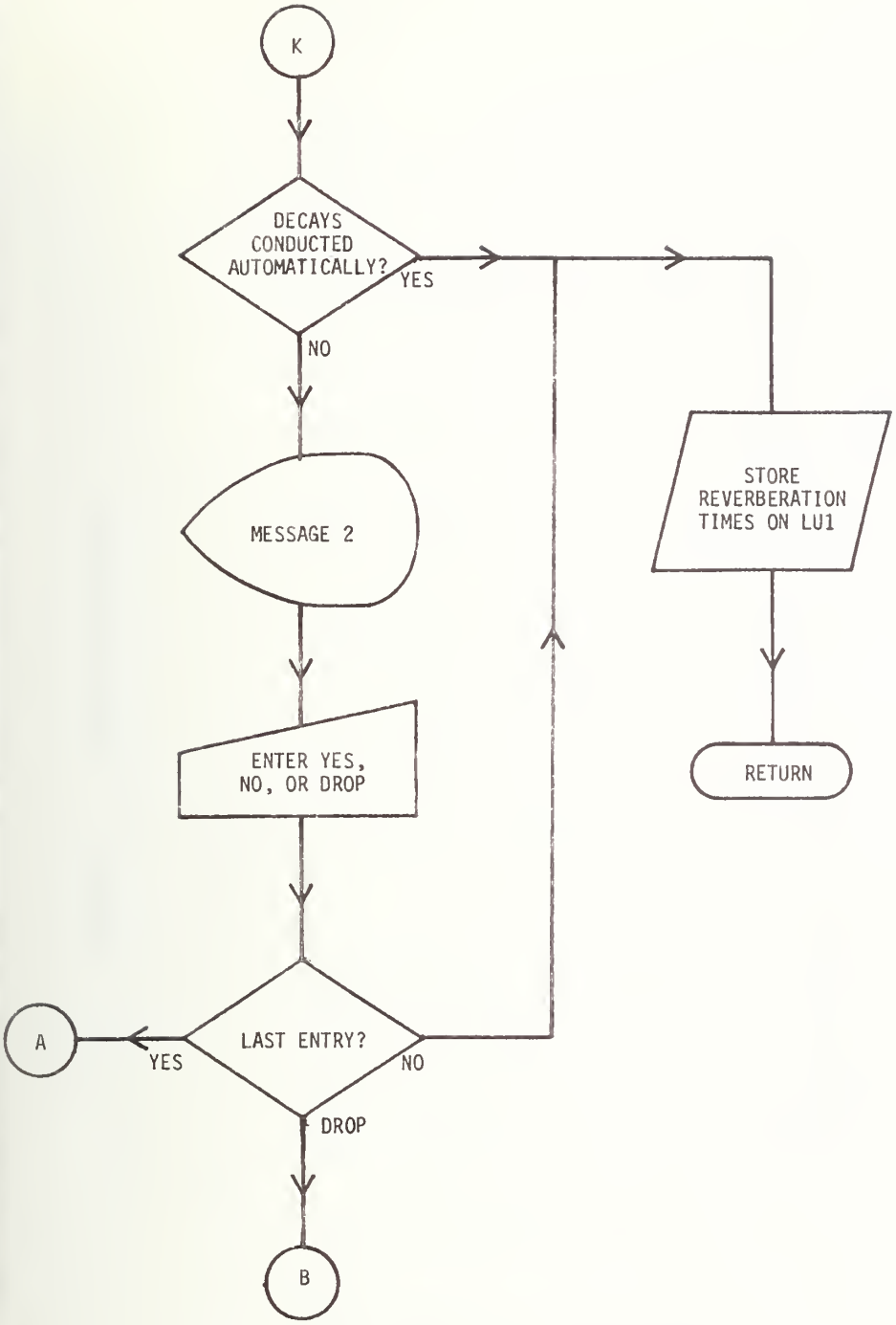
FLOW CHART: SUBROUTINE CURVFT




D
FIND FIRST POINT TO REACH (NOISE + 5 dB): CALL THIS POINT M2
DYNAMIC RANGE < 20 dB?
YES
E
NO
(M2-M1) ≥ 3?
NO
E
YES
F
(1000)
PERFORM LEAST-SQUARES COMPUTATION FOR ONE BAND OVER POINTS M1 to M2
COMPUTE INTERSECTIONS OF LEAST-SQUARES FIT WITH UPPER AND LOWER CURVE-FITTING LIMITS; CALL THESE INTERSECTIONS M3 AND M4, RESPECTIVELY
G
G
|M1-M3| < 1?
YES
NO
M3 ≥ SOUND CUT-OFF POINT OR M1 > SOUND CUT-OFF POINT?
YES
NO
(170)
|M2-M4| < 1?
YES
H
NO
M4 ≤ LAST DATA POINT OR M2 < LAST DATA POINT?
NO
H
YES
(180)
10 ITERATIONS COMPLETED?
YES
I
NO
J


J
M3 < SOUND CUT-OFF POINT?
NO
YES
SET M3 = SOUND CUT-OFF POINT
M4 < LAST DATA POINT?
NO
YES
SET M4 = LAST DATA POINT
(M4-M3) ≥ 3?
NO
YES
SET M1 = M3 M2 = M4
F
I
(190)
|M1-M3|< 2?
YES
NO
M3 ≥ SOUND CUT-OFF POINT OR M1 > SOUND CUT-OFF POINT?
YES
NO
(200)
|M2-M4|< 2?
YES
H
NO
M4 ≤ LAST DATA POINT OR M2 < LAST DATA POINT?
NO
H
YES
E


E
H
PRINT LINE FOR PROPER BAND ON PRINTOUT 1; "*" USED TO INDICATE DECAY CURVE REJECTED
PRINT LINE FOR PROPER BAND ON PRINTOUT 1
EXCLUDE DECAY CURVE FROM SET OF CURVES TO BE AVERAGED
INCLUDE DECAY CURVE IN SET OF CURVES TO BE AVERAGED
COMPLETED WITH LAST BAND?
NO
C
YES
K


K
DECAYS CONDUCTED AUTOMATICALLY?
YES
NO
MESSAGE 2
ENTER YES, NO, OR DROP
LAST ENTRY?
A
YES
NO
DROP
B
STORE REVERBERATION TIMES ON LU1
RETURN

CRT TERMINAL MESSAGES: SUBROUTINE CURVFT

MESSAGE NO.	FORMAT NO.	MESSAGE
1	10	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE FOR CURVE-FITTING PARAMETERS (ENTER "0" IF YOU DO NOT WANT TO PROCESS THIS DECAY)
2	910	DO YOU WANT ANOTHER PRINT OF THESE RESULTS? (ENTER "DROP" IF YOU WANT TO DROP THIS DECAY FROM THE AVERAGES)

PRINTOUT 1 - SUBROUTINE CURVFT

DECAY NO. 1

BAND NO.	FREQ.	REVERB. TIME	SIGMA	SIGNAL LEVEL	NOISE LEVEL	NO. PTS.	M1	M2	NEW M1	NEW M2	DECAY RATE	INTCEPT	RMS OF RESIDUALS	NO. OF REPETITIONS
14	25	23.41	0.22	44.72	10.11	78	45	122	45.53	121.95	2.56	51.15	0.49	2
15	32	18.80	0.65	45.97	8.29	72	48	119	47.43	118.37	3.19	55.81	1.94	2
16	40	12.35	0.73	45.20	5.87	48	41	88	38.64	88.63	4.86	58.51	2.76	2
17	50	9.17	0.52	46.93	8.88	36	51	86	50.32	85.48	6.55	74.27	2.33	2
18	63	5.13	0.49	44.67	6.57	22	42	63	42.66	62.37	11.71	88.52	3.35	2
19	80	3.77	0.38	47.83	8.31	17	43	59	43.33	58.70	15.93	110.38	3.24	3
20	100	4.51	0.41	47.65	7.80	17	41	57	38.56	57.19	13.32	92.75	2.44	2
21	125	4.10	0.23	49.87	6.01	20	42	61	42.06	61.76	14.63	105.05	2.10	2
22	160	4.32	0.17	48.19	0.61	25	45	69	45.61	69.01	13.89	105.27	1.92	2
23	200	5.39	0.19	48.01	0.09	31	44	74	44.06	73.61	11.12	91.00	2.00	2
24	250	6.52	0.18	45.85	0.06	35	45	79	45.05	78.43	9.21	81.48	1.52	2
25	315	6.63	0.20	46.61	0.02	36	46	81	45.91	80.77	9.05	82.32	1.68	2
26	400	6.80	0.15	47.35	0.08	38	46	83	46.27	82.76	8.83	82.37	1.28	2
27	500	6.67	0.13	46.68	0.07	37	46	82	46.39	81.48	9.00	82.59	1.10	2
28	630	6.08	0.11	47.00	0.30	33	46	78	46.00	78.05	9.87	86.51	1.00	2
29	800	5.93	0.09	46.84	0.39	32	46	77	45.89	76.94	10.11	87.32	0.81	2
30	1000	5.80	0.10	46.96	0.45	32	45	76	45.49	75.90	10.35	88.07	0.89	2
31	1250	6.00	0.11	47.56	0.34	33	45	77	45.09	77.23	10.01	86.76	1.01	2
32	1600	5.71	0.08	48.34	0.48	31	45	75	44.44	75.67	10.51	89.06	0.71	2
33	2000	5.34	0.10	47.48	0.61	29	45	73	44.72	73.04	11.24	91.69	0.91	2
34	2500	4.66	0.08	46.09	0.34	25	44	68	44.29	68.14	12.87	96.88	0.77	2
35	3150	4.12	0.05	47.09	0.38	22	44	65	43.30	65.01	14.58	103.87	0.51	2
36	4000	3.37	0.05	45.92	0.26	18	43	60	42.99	60.18	17.81	115.83	0.53	2
37	5000	2.69	0.04	45.95	0.22	15	42	56	42.45	56.23	22.27	133.41	0.60	2
38	6300	2.21	0.03	45.10	0.18	12	42	53	41.82	52.82	27.15	151.10	0.50	2
39	8000	1.57	0.03	46.05	0.24	9	42	50	41.89	49.96	38.10	197.11	0.54	2
40	10000	1.15	0.02	47.02	0.34	7	42	48	41.50	47.58	51.97	252.82	0.53	2
41	12500	0.82	0.02	46.27	0.51	6	41	46	41.43	45.61	73.52	339.02	0.72	2
42	16000	0.55	0.01	45.35	2.28	4	41	44	41.20	43.77	109.06	479.58	0.40	2
*43	20000			29.65	2.51	3	41	43						0
	(HZ)	(SEC)	(SEC)	(DB)	(DB)						(DB/SEC)	(DB)	(DB)	

SUBROUTINE CURVFT

```
$ASSM
CURVFT PROG CURVFT - SUBROUTINE FOR LEAST-SQUARES ANALYSIS OF DECAYS
$FORT
C    ------------------------------
C    OTHER SUBROUTINES CALLED:  BAND
C    ------------------------------
       SUBROUTINE CURVFT(N,NSIG,X,X0,S0,S1,ARRAY1,ARRAY2,DELTA,
      1FREQ,NDCAYS,AUTO,LU,BLOW,BHIGH,CUPPER,CLOWER,MM1,MM2,MXRDG)
       INTEGER*2 B,NSIG,N,X0(6060),FREQ(30),LU,FF,AUTO,
      1BLOW,BHIGH,MM1(30),MM2(30)
       INTEGER*4 RETRN
       DIMENSION X(200),S0(30,12),S1(30),ARRAY1(30),ARRAY2(30)
       DATA FF/X'0C00'/
       DO 2 I=1,30
       ARRAY1(I)=0.
2      CONTINUE
       IF (AUTO.NE.0) GO TO 18
5      WRITE (5,10)
10     FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE FOR CURVE-
      1FITTING PARAMETERS'/'(ENTER "0" IF YOU DO NOT WANT TO
      2 PROCESS THIS DECAY)')
       READ (5,15)LU
15     FORMAT (I1)
       IF (LU.GT.0) GO TO 18
16     NDCAYS=NDCAYS-1
       GO TO 2020
18     ASSIGN 150 TO RETRN
       WRITE (LU,19)NDCAYS
19     FORMAT (X,X/X/45X,'DECAY NO.',I4)
       IF (LU.EQ.5) GO TO 22
       WRITE (LU,20)
20     FORMAT (X/X/X,'BAND',9X,'REVERB.',8X,'SIGNAL',3X,'NOISE',3X,
      1'NO.',13X,'NEW',5X,'NEW',4X,'DECAY',12X,'RMS OF',6X,'NO. OF'/
      2X,'NO.',2X,'FREQ.',3X,'TIME',3X,'SIGMA',2(3X,'LEVEL'),2X,'PTS.',
      32X,'M1',3X,'M2',5X,'M1',6X,'M2',5X,'RATE',2X,'INTCEPT',2X,
      4'RESIDUALS',2X,'REPETITIONS'/X)
       GO TO 25
22     WRITE (LU,23)
23     FORMAT (X/X/X/'BAND',9X,'REVERB.',8X,'SIGNAL',3X,'NOISE',
      13X,'NO.',13X,'NEW',5X,'NEW'/
      2X,'NO.',2X,'FREQ.',3X,'TIME',3X,'SIGMA',2(3X,'LEVEL'),
      32X,'PTS.',2X,'M1',3X,'M2',5X,'M1',6X,'M2'/X)
25     DO 900 B=BLOW,BHIGH
       IB=B-13
       CALL BAND(B,N,X,X0,6060)
       SUM1=0
       DO 45 I=1,NSIG
       SUM1=SUM1+10.**(X(I)/10)
45     CONTINUE
       SUM1=SUM1/NSIG
       S1(IB)=10*ALOG10(SUM1)
       NREP=0
       M1=NSIG+1
       DO 130 I=M1,N
       IF (X(I)-S0(IB,MXRDG)-5.)131,131,130
130    CONTINUE
131    M2=I
       IF ((S1(IB)-S0(IB,MXRDG)).LT.20.) GO TO 850
       IF (M2-M1-3)850,1000,1000
150    NREP=NREP+1
```

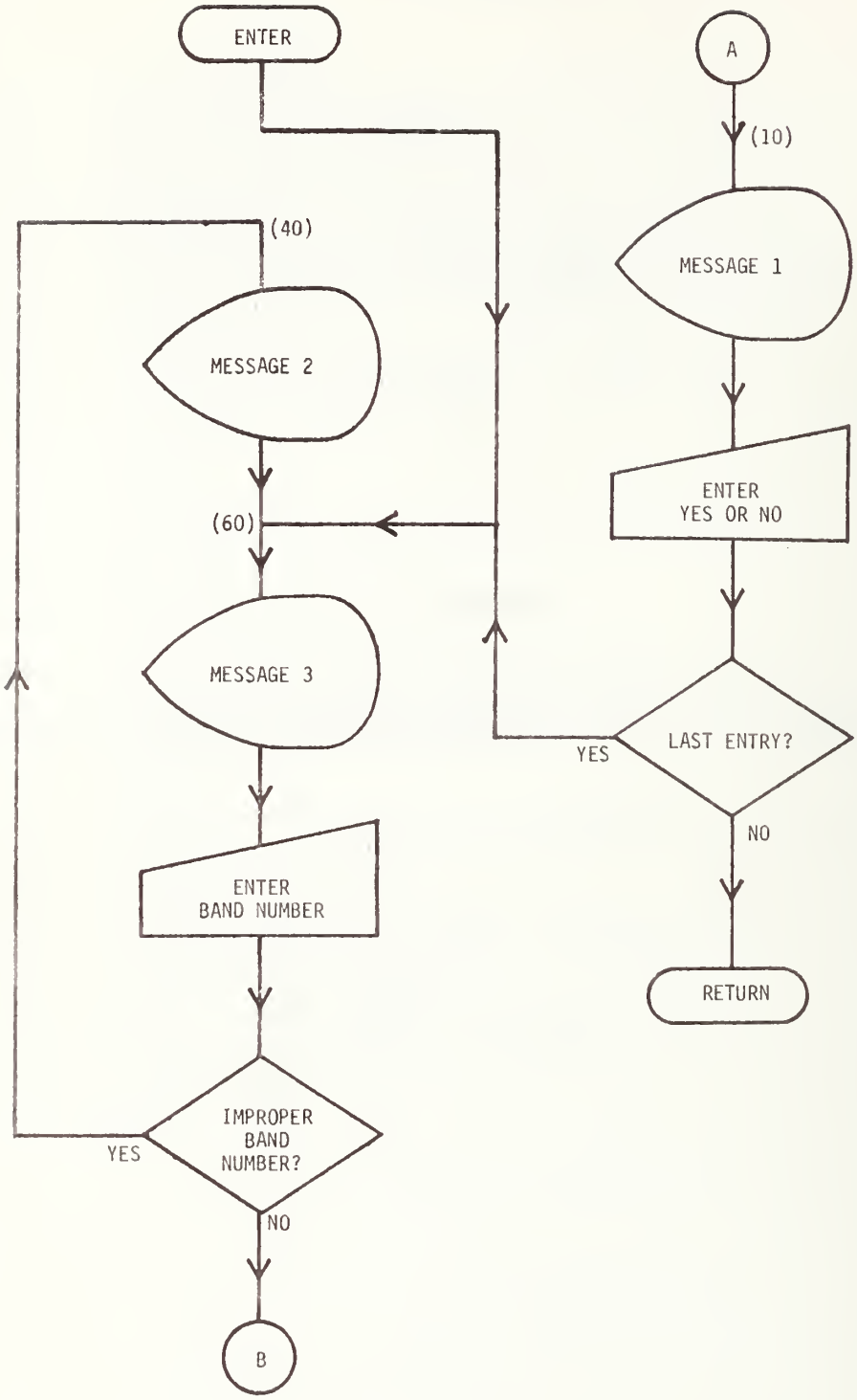
```
      AXIS1=(SLOPE*(M1-1)*DELTA)+AXIS0
      RM3=1+(AXIS1-S1(IB)+CUPPER)/(SLOPE*DELTA)
      RM4=1+(AXIS1-S0(IB,MXRDG)-CLOWER)/(SLOPE*DELTA)
      M3=INT(RM3+.5)
      M4=INT(RM4+.5)
      IF (ABS(M1-RM3).LT.1.) GO TO 170
      IF ((M3.GE.NSIG+1).OR.(M1.GT.NSIG+1)) GO TO 180
170   IF (ABS(M2-RM4).LT.1.)GO TO 880
      IF ((M4.LE.N).OR.(M2.LT.N)) GO TO 180
      GO TO 880
180   IF (NREP.EQ.10) GO TO 190
      IF (M3.LT.NSIG+1) M3=NSIG+1
      IF (M4.GT.N) M4=N
      IF (M4.LT.M3+3) GO TO 860
      M1=M3
      M2=M4
      GO TO 1000
190   IF (ABS(M1-RM3).LT.2.) GO TO 200
      IF ((M3.GE.NSIG+1).OR.(M1.GT.NSIG+1)) GO TO 860
200   IF (ABS(M2-RM4).LT.2.) GO TO 880
      IF ((M4.LE.N).OR.(M2.LT.N)) GO TO 860
      GO TO 880
850   M=M2-M1+1
      IF (LU.EQ.5) GO TO 854
      WRITE (LU,852)B,FREQ(IB),S1(IB),S0(IB,MXRDG),M,M1,M2,NREP
852   FORMAT ('*',I2,2X,I5,18X,2(F6.2,2X),3(I3,2X),48X,I2)
      GO TO 900
854   WRITE (LU,856) B,FREQ(IB),S1(IB),S0(IB,MXRDG),M,M1,M2
856   FORMAT ('*',I2,2X,I5,18X,2(F6.2,2X),3(I3,2X))
      GO TO 900
860   IF (LU.EQ.5) GO TO 864
      WRITE (LU,862)B,FREQ(IB),T60,VT60,S1(IB),S0(IB,MXRDG),
     1M,M1,M2,RM3,RM4,SLOPE,AXIS1,VRMS,NREP
862   FORMAT ('*',I2,2X,I5,2X,4(F6.2,2X),3(I3,2X),4(F6.2,2X),X,
     1F6.2,9X,I2)
      GO TO 900
864   WRITE (LU,866) B,FREQ(IB),T60,VT60,S1(IB),S0(IB,MXRDG),
     1M,M1,M2,RM3,RM4
866   FORMAT ('*',I2,2X,I5,2X,4(F6.2,2X),3(I3,2X),2(F6.2,2X))
      GO TO 900
870   IF (LU.EQ.5) GO TO 854
      WRITE (LU,872)B,FREQ(IB),S1(IB),S0(IB,MXRDG),M,M1,M2,
     1SLOPE,NREP
872   FORMAT ('*',I2,2X,I5,18X,2(F6.2,2X),3(I3,2X),16X,F6.2,18X,I2)
      GO TO 900
880   IF (LU.EQ.5) GO TO 884
      WRITE (LU,882)B,FREQ(IB),T60,VT60,S1(IB),S0(IB,MXRDG),
     1M,M1,M2,RM3,RM4,SLOPE,AXIS1,VRMS,NREP
882   FORMAT (X,I2,2X,I5,2X,4(F6.2,2X),3(I3,2X),4(F6.2,2X),X,
     1F6.2,9X,I2)
      GO TO 888
884   WRITE (LU,886) B,FREQ(IB),T60,VT60,S1(IB),S0(IB,MXRDG),
     1M,M1,M2,RM3,RM4
886   FORMAT (X,I2,2X,I5,2X,4(F6.2,2X),3(I3,2X),2(F6.2,2X))
888   ARRAY1(IB)=T60
      ARRAY2(IB)=AXIS1
      MM1(IB)=M1
      MM2(IB)=M2
900   CONTINUE
      IF (LU.EQ.5) GO TO 906
```

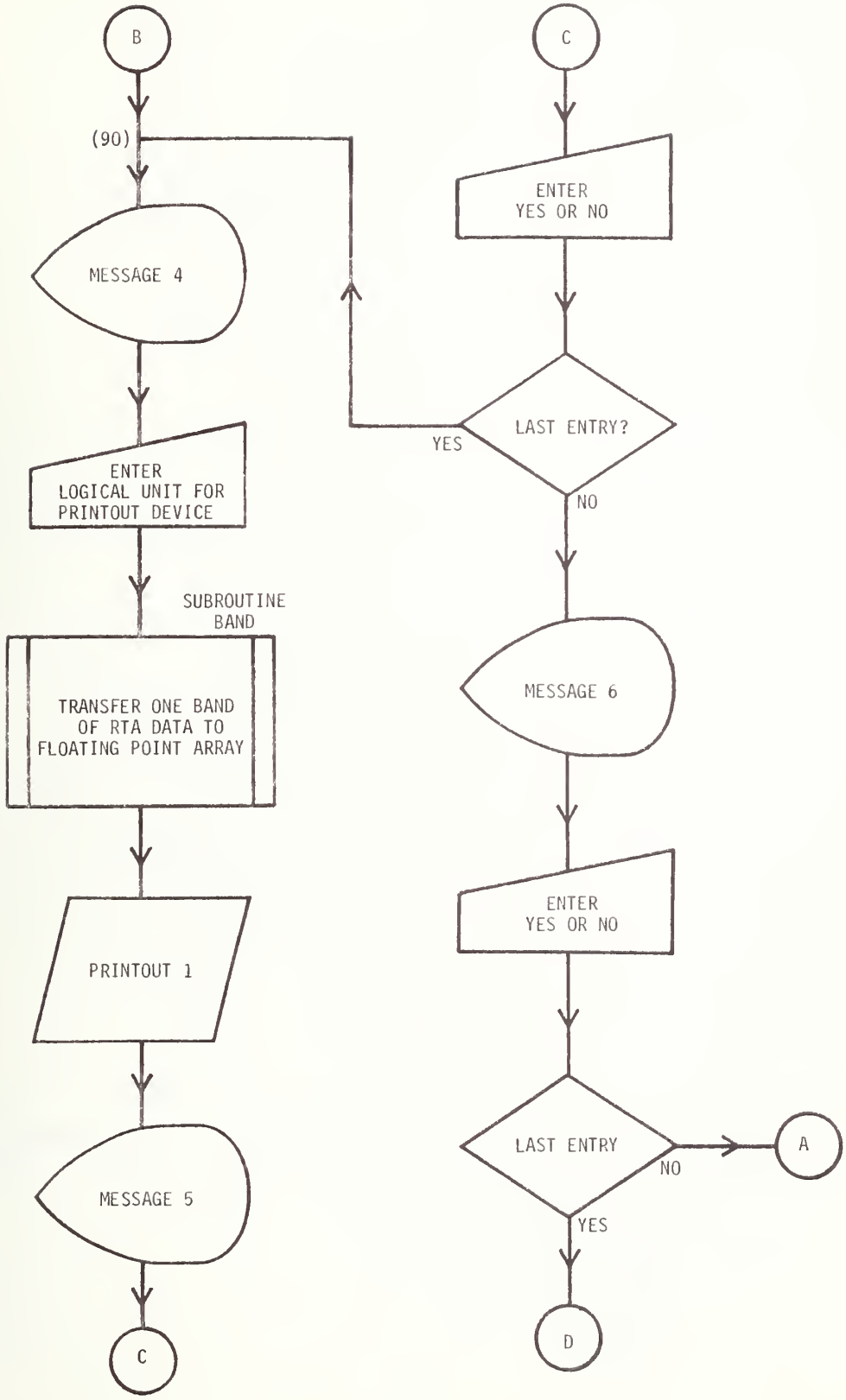
```
121         WRITE (LU,905)FF
122 905     FORMAT (X/6X,'(HZ)',2(3X,'(SEC)'),2(4X,'(DB)'),3X,'(DB/SEC)',
123        13X,'(DB)',5X,'(DB)'/A1)
124         GO TO 908
125 906     WRITE (LU,907) FF
126 907     FORMAT (X/6X,'(HZ)',2(3X,'(SEC)'),2(4X,'(DB)')/A1)
127 908     IF (AUTO.NE.0) GO TO 2000
128         WRITE (5,910)
129 910     FORMAT ('DO YOU WANT ANOTHER PRINT OF THESE RESULTS?'/
130        1'(ENTER "DROP" IF YOU WANT TO DROP THIS DECAY FROM THE
131        2 AVERAGES)')
132         READ (5,915)GS
133 915     FORMAT (A4)
134         IF (GS.EQ.'DROP') GO TO 16
135         IF (GS.EQ.'NO')GO TO 2000
136         GO TO 5
137 1000    M=M2-M1+1
138         SUM1=0
139         SUM2=0
140         DO 1050 I=1,M
141         J=M1+I-1
142         SUM1=SUM1+X(J)
143         SUM2=SUM2+(I-1)*X(J)
144 1050    CONTINUE
145         SUM2=SUM2*DELTA
146         CON1=DELTA*M*(M-1)/2
147         CON2=DELTA*DELTA*M*(M-1)*((2*M)-1)/6
148         AXIS0=(SUM2-(SUM1*CON2/CON1))/(CON1-(M*CON2/CON1))
149         SLOPE=((M*AXIS0)-SUM1)/CON1
150         IF (SLOPE.LT..1)GO TO 870
151         T60=60./SLOPE
152         SUM1=0
153         DO 1100 I=1,M
154         J=M1+I-1
155         CON3=X(J)+(SLOPE*(I-1)*DELTA)-AXIS0
156         SUM1=SUM1+(CON3*CON3)
157 1100    CONTINUE
158         VRMS=SQRT(SUM1/M)
159         VSLOPE=VRMS*SQRT(M/((M*CON2)-(CON1*CON1)))
160         VAXIS0=VRMS*SQRT(CON2/M)
161         VT60=T60*VSLOPE/SLOPE
162         GO TO RETRN
163 2000    WRITE (1,2010)(ARRAY1(I),I=1,15)
164         WRITE (1,2010)(ARRAY1(I),I=16,30)
165 2010    FORMAT (15F8.4)
166 2020    RETURN
167         END
```

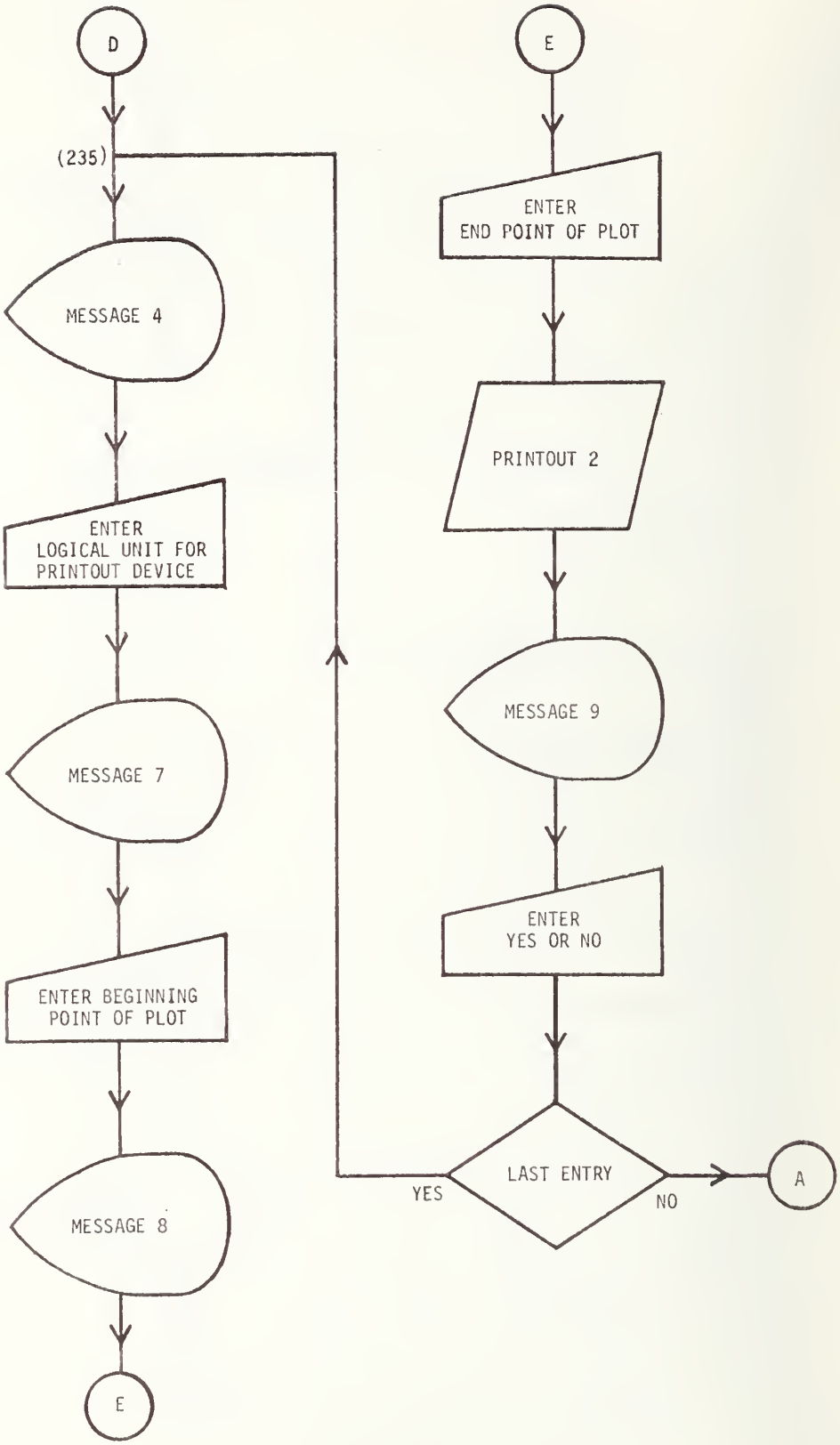
APPENDIX E

Subroutine DSDATA Flow Chart, Terminal Messages, Printouts, and Listings

FLOW CHART: SUBROUTINE DSDATA




B
(90)
MESSAGE 4
ENTER
LOGICAL UNIT FOR
PRINTOUT DEVICE
SUBROUTINE
BAND
TRANSFER ONE BAND
OF RTA DATA TO
FLOATING POINT ARRAY
PRINTOUT 1
MESSAGE 5
C
C
ENTER
YES OR NO
LAST ENTRY?
YES
NO
MESSAGE 6
ENTER
YES OR NO
LAST ENTRY
NO
A
YES
D


D
(235)
MESSAGE 4
ENTER
LOGICAL UNIT FOR
PRINTOUT DEVICE
MESSAGE 7
ENTER BEGINNING
POINT OF PLOT
MESSAGE 8
E
E
ENTER
END POINT OF PLOT
PRINTOUT 2
MESSAGE 9
ENTER
YES OR NO
LAST ENTRY
YES
NO
A

CRT TERMINAL MESSAGES: SUBROUTINE DSDATA

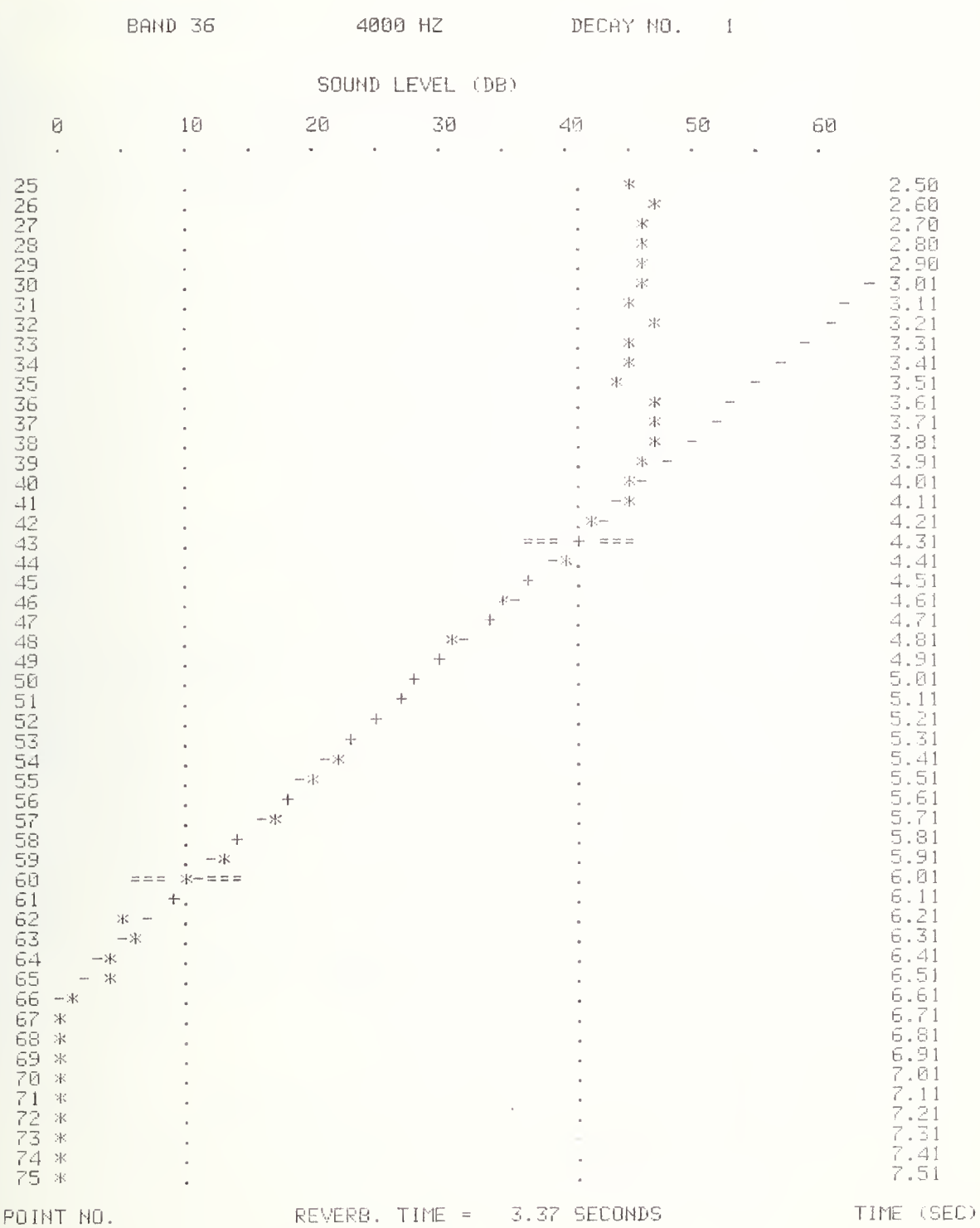
MESSAGE NO.	FORMAT NO.	MESSAGE
1	20	DO YOU WANT TO SEE ANOTHER BAND OF DATA?
2	50	YOU HAVE ENTERED AN IMPROPER BAND NUMBER
3	70	ENTER BAND NUMBER
4	100	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE
5	200	DO YOU WANT ANOTHER PRINT OF THIS TABLE?
6	230	DO YOU WANT A PLOT OF BAND 30?
7	238	ENTER BEGINNING POINT OF DESIRED PLOT (3 DIGITS)
8	240	ENTER END POINT OF DESIRED PLOT (3 DIGITS)
9	260	DO YOU WANT ANOTHER PLOT OF BAND 30?

PRINTOUT 1 - SUBROUTINE DSDATA

BAND 36 4000 HZ DECAY NO. 1

1	2	3	4	5	6	7	8	9	10	
46.00	46.25	46.25	45.25	46.50	46.25	45.75	46.00	46.25	45.75	1
45.50	47.25	46.75	45.25	46.00	47.00	45.25	46.00	45.25	45.75	2
46.00	44.75	44.75	45.50	45.25	46.75	46.25	45.50	45.75	46.25	3
44.75	47.00	45.00	45.00	43.75	47.25	46.50	46.75	46.25	45.00	4
44.75	42.25	40.75	40.25	37.00	35.25	34.25	31.25	29.75	28.00	5
26.75	25.25	22.50	21.50	19.75	17.75	16.75	13.50	13.00	10.00	6
9.00	5.25	6.25	4.25	3.50	0.50	0.00	0.00	0.00	0.00	7
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	13
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	17
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	19
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	20

PRINTOUT 2 - SUBROUTINE DSDATA



SUBROUTINE DSDATA

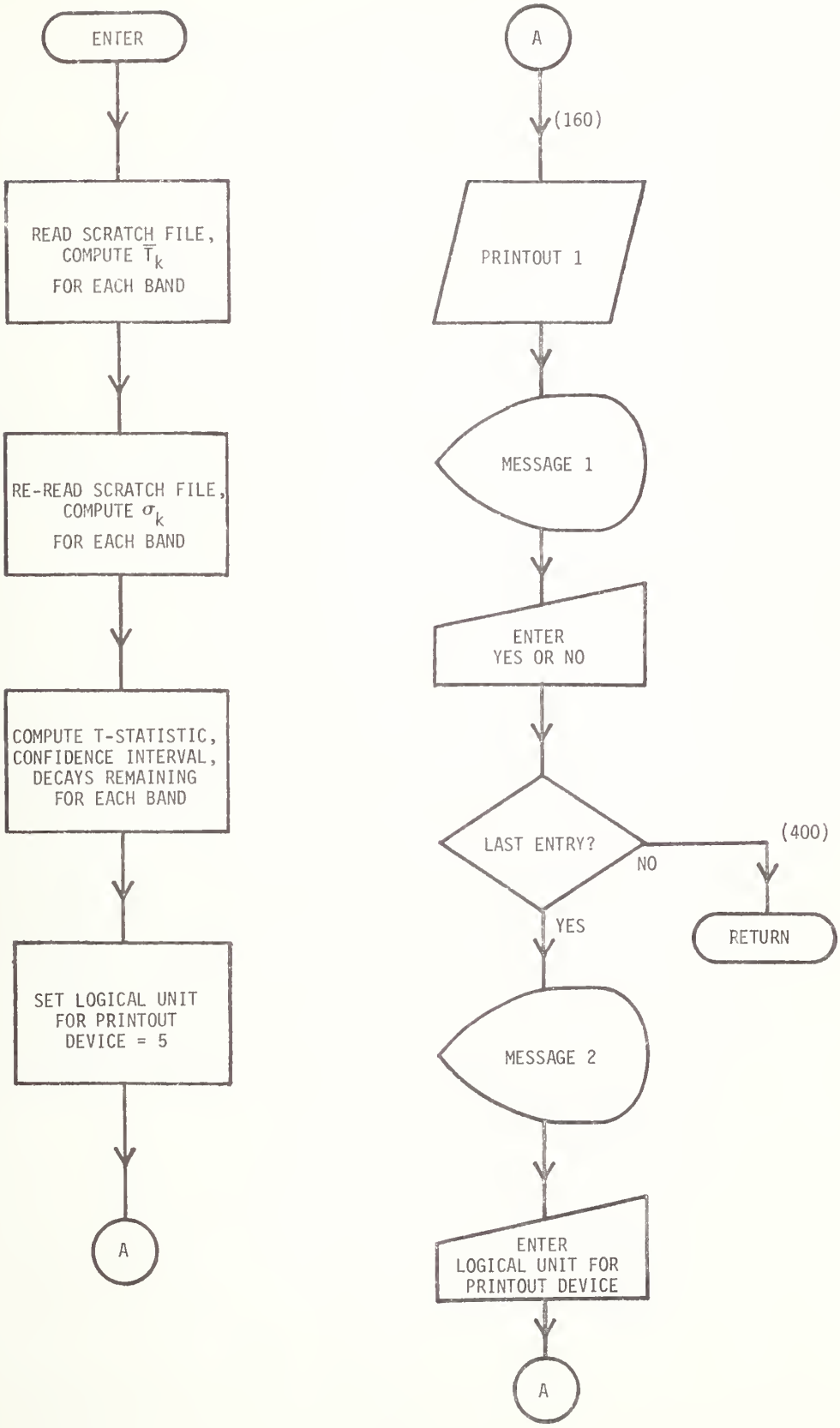
```
$ASSM
DSDATA PROG DSDATA - SUBROUTINE FOR DISPLAYING DECAY DATA POINTS
$FORT
C    ------------------------------
C    OTHER SUBROUTINES CALLED:  BAND
C    ------------------------------
      SUBROUTINE DSDATA(X,X0,ARRAY1,ARRAY2,FREQ,N,NDCAYS,LINE,DELTA,
     1S0,S1,CUPPER,CLOWER,MM1,MM2,MXPDG)
      INTEGER*2 B,LU,N,X0(6060),FREQ(30),LINE(65),MM1(30),MM2(30)
      INTEGER*2 BLANK,STAR,PLUS,MINUS,DOT,EQUALS,FF
      DIMENSION X(200),ARRAY1(30),ARRAY2(30),S0(30,12),S1(30)
      DATA BLANK,STAR,PLUS,MINUS,DOT,EQUALS,FF/X'2000',X'2A00',
     1X'2B00',X'2D00',X'2E00',X'3D00',X'0C00'/
      GO TO 60
10    WRITE (5,20)
20    FORMAT ('DO YOU WANT TO SEE ANOTHER BAND OF DATA?')
      READ (5,30) GS
30    FORMAT (A4)
      IF (GS.EQ.'NO')GO TO 900
      GO TO 60
40    WRITE (5,50)
50    FORMAT ('YOU HAVE ENTEPED AN IMPROPER BAND NUMBER')
60    WRITE (5,70)
70    FORMAT ('ENTER BAND NUMBER')
      READ (5,80)B
80    FORMAT (I2)
      IF ((B.LT.14).OR.(B.GT.43)) GO TO 40
90    WRITE (5,100)
100   FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE')
      READ (5,110)LU
110   FORMAT (I1)
      IB=B-13
      CALL BAND(B,N,X,X0,6060)
      WRITE (LU,130)B,FREQ(IB),NDCAYS,(I,I=1,10)
130   FORMAT (3X,3X,3X,10X,'BAND ',I2,10X,I5,' HZ',10X,'DECAY NO.',I4/
     1(3X,3X,3X,10(I2,5X)/3X)
      NLINES=N/10
      DO 150 I=1,NLINES
      J1=(I-1)*10+1
      J2=(I-1)*10+10
      WRITE (LU,140)(X(J),J=J1,J2),I
140   FORMAT (10(F6.2,3X),I2)
150   CONTINUE
      WRITE (LU,160)FF
160   FORMAT (A1)
      WRITE (5,200)
200   FORMAT ('DO YOU WANT ANOTHER PRINT OF THIS TABLE?')
      READ (5,30) GS
      IF (GS.NE.'NO')GO TO 90
      WRITE (5,230)B
230   FORMAT ('DO YOU WANT A PLOT OF BAND ',I2,' ?')
      READ (5,30)GS
      IF (GS.EQ.'NO') GO TO 10
235   WRITE (5,100)
      READ (5,110)LU
      WRITE (5,238)
238   FORMAT ('ENTER BEGINNING POINT OF DESIRED PLOT    (3 DIGITS)')
      READ (5,250)M1
      WRITE (5,240)
240   FORMAT ('ENTER END POINT OF DESIRED PLOT    (3 DIGITS)')
```

```
      READ (5,250)N2
250   FORMAT (I3)
      IF (ARRAY1(IB).LT.0.01) GO TO 370
      SLOPE=60./ARRAY1(IB)
370   WRITE (LU,380)B,FREQ(IB),NDCAYS
380   FORMAT (10X,'BAND ',I2,10X,I5,' HZ',10X,'DECAY NO.',I4/X,
     1/25X,'SOUND LEVEL (DB)'/X/4X,'0',9X,'10',8X,'20',8X,'30',8X,
     2'40',8X,'50',8X,'60'/4X,13('.',4X)/X)
      J3=S1(IB)-CUPPER+1.5
      J4=S0(IB,MXRDG)+CLOWER+1.5
390   DO 500 I=N1,N2
      DO 400 J=1,65
      LINE(J)=BLANK
400   CONTINUE
      TIME=I*DELTA
      J1=X(I)+1.5
      IF (ARRAY1(IB).LT.0.01) GO TO 405
      LINE(J3)=DOT
      LINE(J4)=DOT
      IF ((I.NE.MM1(IB)).AND.(I.NE.MM2(IB))) GO TO 404
      J5=J1-4
      J6=J1-2
      DO 402 J=J5,J6
      LINE(J)=EQUALS
402   CONTINUE
      J5=J1+2
      J6=J1+4
      DO 403 J=J5,J6
      LINE(J)=EQUALS
403   CONTINUE
404   J2=1.5+ARRAY2(IB)-SLOPE*(I-1)*DELTA
      IF ((J2.LT.1).OR.(J2.GT.65)) GO TO 405
      IF (I2.EQ.J1) GO TO 410
      LINE(J2)=MINUS
405   LINE(J1)=STAR
      GO TO 420
410   LINE(J1)=PLUS
420   WRITE (LU,430)I,(LINE(J),J=1,65),TIME
430   FORMAT (I3,X,65A1,F5.2)
500   CONTINUE
      IF (ARRAY1(IB).LT.0.01) GO TO 520
      WRITE (LU,510) ARRAY1(IB),FF
510   FORMAT (X/'POINT NO.',T24,'REVERB. TIME = ',F6.2,
     1' SECONDS',T68,'TIME (SEC)'/A1)
      GO TO 600
520   WRITE (LU,530) FF
530   FORMAT (X/'POINT NO.',T68,'TIME (SEC)'/A1)
600   WRITE (5,610) B
610   FORMAT ('DO YOU WANT ANOTHER PLOT OF BAND ',I2,' ?')
      READ (5,30)G5
      IF (G5.EQ.'NO') GO TO 10
      GO TO 235
900   RETURN
      END
```

APPENDIX F

Subroutine AVRGE Flow Chart, Terminal Messages, Printout, and Listings

FLOW CHART: SUBROUTINE AVRGE



CRT TERMINAL MESSAGES: SUBROUTINE AVRGE

MESSAGE NO.	FORMAT NO.	MESSAGE
1	320	DO YOU WANT ANOTHER PRINT OF THESE RESULTS?
2	340	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE

PRINTOUT 1 - SUBROUTINE AVRGE

NUMBER OF DECAYS CONDUCTED = 24

BAND NO.	FREQ. (HZ)	REVERB TIME (SEC)	STD.DEV. (SEC)	CONFIDENCE INTERVAL (%,+OR-)	NO. OF DECAYS	DECAYS REMAINING
14	25	22.11	3.66	7.7	20	262
15	32	17.82	4.02	10.3	21	500
16	40	14.85	3.53	10.0	24	550
17	50	9.51	2.55	11.6	23	709
18	63	4.14	0.57	5.8	24	166
19	80	4.57	0.58	5.3	24	137
20	100	3.02	0.45	6.2	24	197
21	125	3.71	0.35	4.0	24	67
22	160	4.51	0.26	2.4	24	9
23	200	5.19	0.21	1.7	24	0
24	250	6.31	0.17	1.2	24	0
25	315	6.80	0.15	1.0	24	0
26	400	6.97	0.17	1.0	24	0
27	500	6.76	0.15	0.9	24	0
28	630	6.10	0.16	1.1	24	0
29	800	5.91	0.11	0.8	24	0
30	1000	5.94	0.08	0.6	24	0
31	1250	5.92	0.07	0.5	24	0
32	1600	5.66	0.07	0.5	24	0
33	2000	5.19	0.09	0.8	24	0
34	2500	4.63	0.07	0.6	24	0
35	3150	4.01	0.06	0.7	24	0
36	4000	3.34	0.05	0.6	24	0
37	5020	2.70	0.04	0.6	24	0
38	6300	2.12	0.05	0.9	24	0
39	8000	1.58	0.02	0.7	24	0
40	10000	1.16	0.03	1.0	24	0
41	12500	0.84	0.03	1.3	24	0
42	16000	0.61	0.04	2.8	24	19
43	20000				0	

SUBROUTINE AVRGE

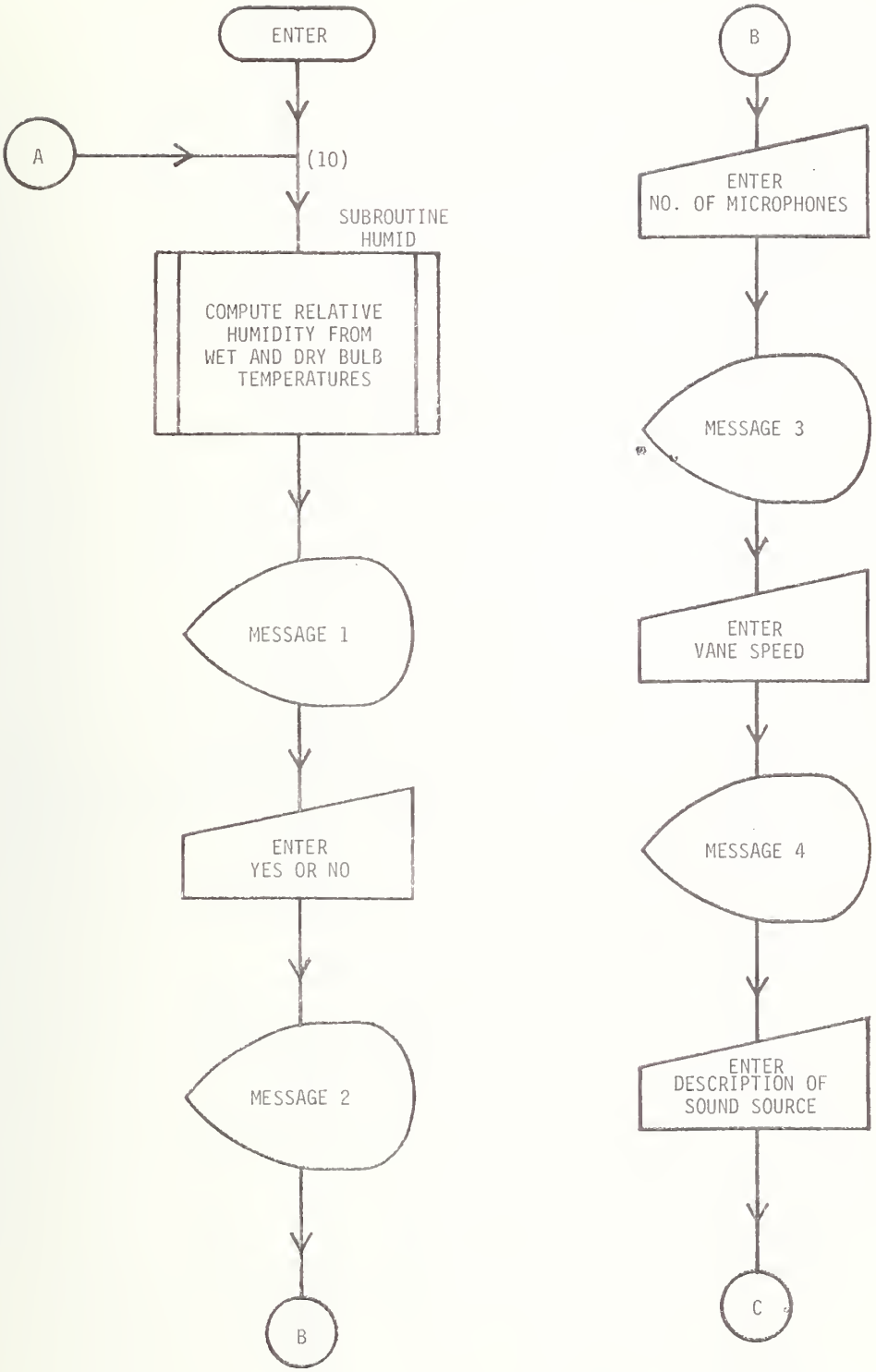
```
$ASSM
AVRGE PROG AVRGE - SUBROUTINE FOR AVERAGING THE RESULTS OF MANY DECAYS
$FORT
C   ------------------------------
C   SUBPROGRAMS CALLED:  TINORM,STUDIN
C   ------------------------------
      SUBROUTINE AVRGE(X,X0,ARRAY1,ARRAY2,NDCAYS,CINTER,CLEVEL,
     1FREQ,BLOW,BHIGH)
      INTEGER*2 X0(60),FREQ(30),B,BLOW,BHIGH,B1,B2,LU,FF
      DIMENSION X(200),ARRAY1(30),ARRAY2(30)
      DATA FF/X'0C00'/
      ALPHA=1.-CLEVEL/100.
      DO 10 I=1,30
      X(I)=0.
      X0(I)=0
10    CONTINUE
      REWIND 1
      DO 20 J=1,NDCAYS
      READ (1,15)(ARRAY1(I),I=1,15)
      READ (1,15)(ARRAY1(I),I=16,30)
15    FORMAT (15F8.4)
      DO 20 I=1,30
      IF (ARRAY1(I).EQ.0.) GO TO 20
      X(I)=X(I)+ARRAY1(I)
      X0(I)=X0(I)+1
20    CONTINUE
      DO 30 I=1,30
      IF (X0(I).EQ.0) GO TO 30
      ARRAY1(I)=X(I)/X0(I)
30    CONTINUE
      DO 40 I=1,30
      X(I)=0.
40    CONTINUE
      REWIND 1
      DO 50 J=1,NDCAYS
      READ (1,15)(ARRAY2(I),I=1,15)
      READ (1,15)(ARRAY2(I),I=16,30)
      DO 50 I=1,30
      IF (ARRAY2(I).EQ.0.) GO TO 50
      X(I)=X(I)+(ARRAY1(I)-ARRAY2(I))*(ARRAY1(I)-ARRAY2(I))
50    CONTINUE
      OPDIN=TINORM(1.-ALPHA/2.)
      NDF=0
      DO 100 I=1,30
      IF (X0(I).LT.2) GO TO 100
      IF (NDF.EQ.(X0(I)-1)) GO TO 70
      NDF=X0(I)-1
      TALPHA=STUDIN(ALPHA,NDF,4.5)
70    ARRAY2(I)=SQRT(X(I)/NDF)
      X(I+30)=100.*ARRAY2(I)*TALPHA/(ARRAY1(I)*SQRT(1.*X0(I)))
      IF ((X(I+30).GT.CINTER) GO TO 80
      X0(I+30)=0
      GO TO 100
80    C=(TALPHA*OPDIN/2.
      C=100.*(C*ARRAY2(I)/(ARRAY1(I)*CINTER)
      C=C*(C+.5-X0(I))
      IF (C.LE.32767.) GO TO 90
      X0(I+30)=32,67
      GO TO 100
90    X0(I+30)=C
```

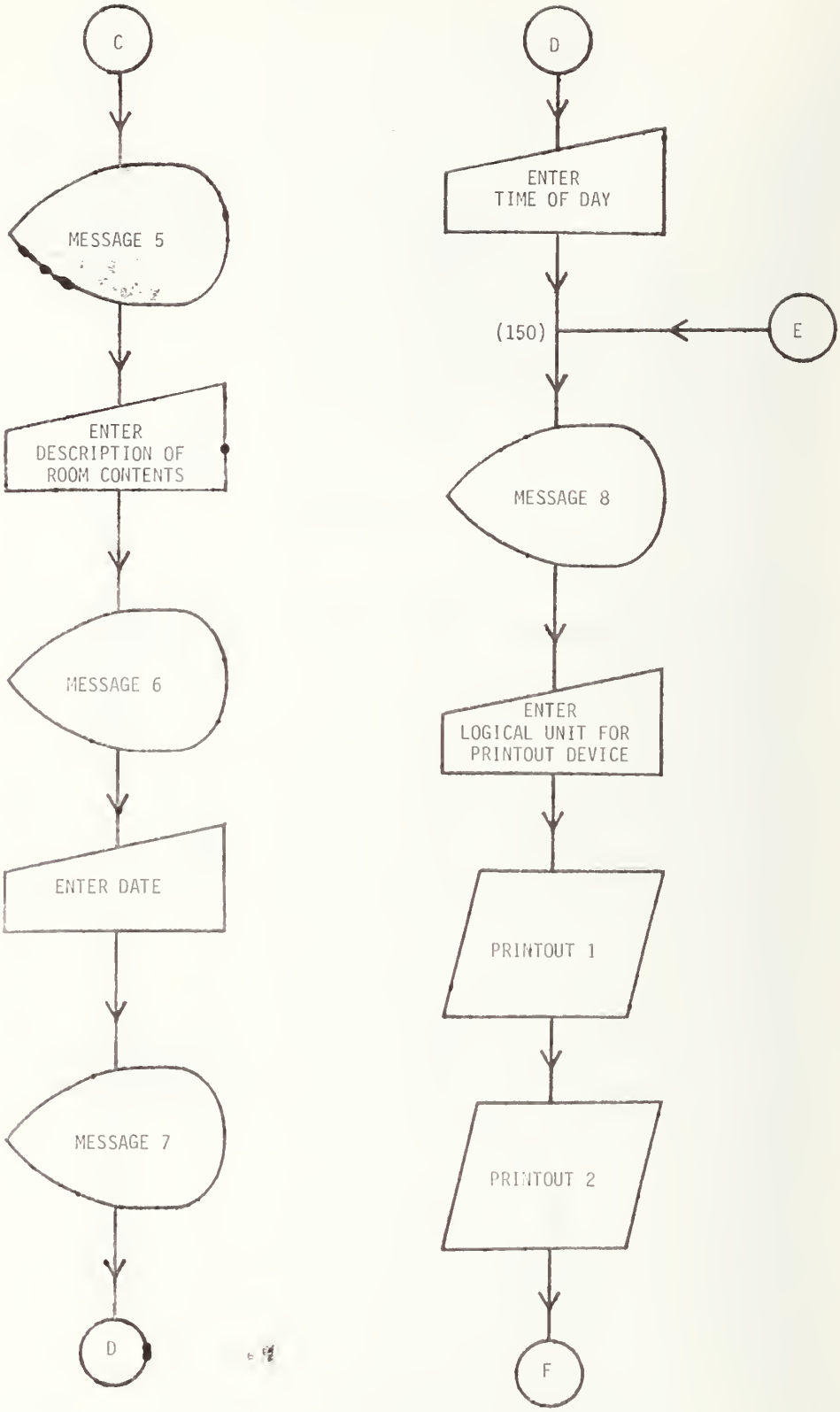
```
      IF (X0(I+30).LT.1) X0(I+30)=1
100   CONTINUE
      LU=5
160   WRITE (LU,170)NDCAYS
170   FORMAT ('1'//T11,'NUMBER OF DECAYS CONDUCTED =',I4//X//X
     1'BAND',T14,'RESERR',T31,'CONFIDENCE',T43,
     2'NO. OF',T52,'DECAYS'/
     312,'NO.',T7,'FREQ.',T15,'TIME',T22,'STD.DEV.',T32,'INTERVAL',
     4T43,'DECAYS',T51,'REMAINING'/
     5X,'(HZ)',3X,'(SEC)',4X,'(SEC)',T32,'(+OR-)'/X)
      DO 300 B=BLOW,BHIGH
      I=B-13
      IF (X0(I).EQ.0) GO TO 220
      IF (X0(I).EQ.1) GO TO 240
      WRITE (LU,200)B,FREQ(I),ARRAY1(I),ARRAY2(I),X(I+30),X0(I),X0(I+30)
200   FORMAT (/,I2,2X,I5,3X,F6.2,3X,F6.2,5X,F5.1,5X,I4,5X,I5)
      GO TO 300
220   ARRAY1(I)=0.
      ARRAY2(I)=0.
      WRITE (LU,230)B,FREQ(I)
230   FORMAT (/,I2,2X,I5,35X,'0')
      GO TO 300
240   ARRAY2(I)=0.
      WRITE (LU,250)B,FREQ(I),ARRAY1(I)
250   FORMAT (/,I2,2X,I5,3X,F6.2,27X,'1')
300   CONTINUE
      WRITE (LU,310) FF
310   FORMAT (A1)
      WRITE (5,320)
320   FORMAT ('DO YOU WANT ANOTHER PRINT OF THESE RESULTS?')
      READ (5,330)GS
330   FORMAT (A4)
      IF (GS.EQ.'NO') GO TO 400
      WRITE (5,340)
340   FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE')
      READ (5,350)LU
350   FORMAT (I1)
      GO TO 160
400   RETURN
      END
```

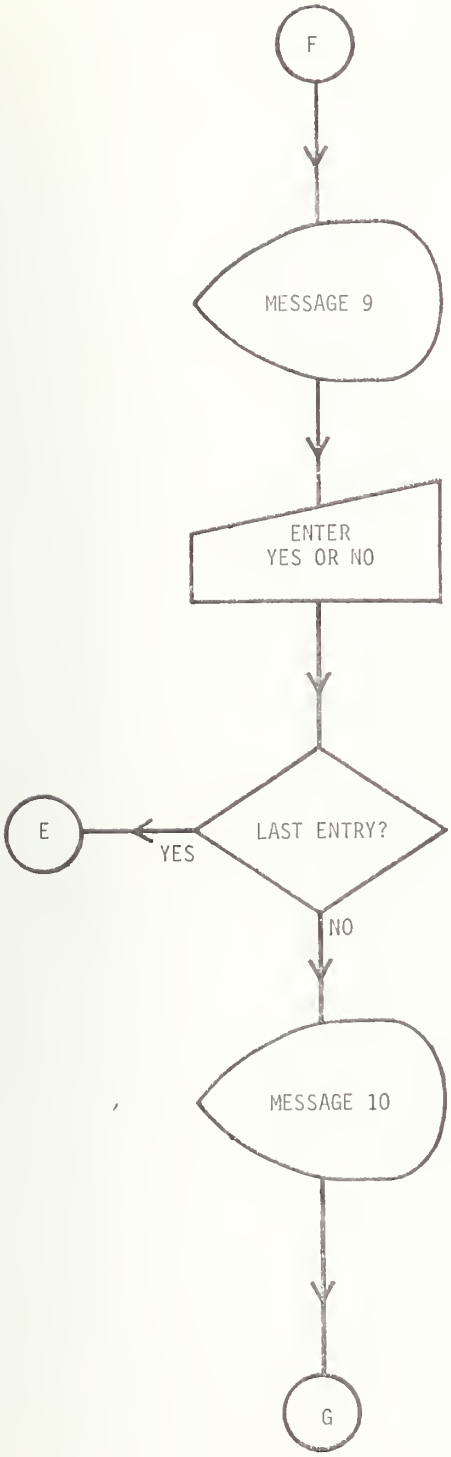
APPENDIX G

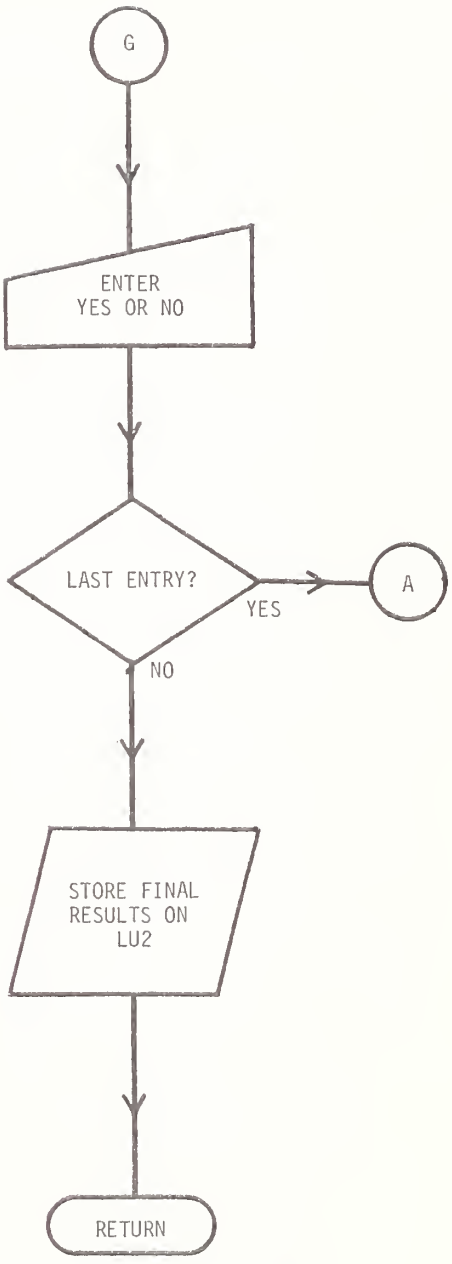
Subroutine RSULTS Flow Chart, Terminal Messages, Printouts and Listings

FLOW CHART: SUBROUTINE RSULTS




C
MESSAGE 5
ENTER
DESCRIPTION OF
ROOM CONTENTS
MESSAGE 6
ENTER DATE
MESSAGE 7
D
D
ENTER
TIME OF DAY
(150)
E
MESSAGE 8
ENTER
LOGICAL UNIT FOR
PRINTOUT DEVICE
PRINTOUT 1
PRINTOUT 2
F


F
MESSAGE 9
ENTER
YES OR NO
E
YES
LAST ENTRY?
NO
MESSAGE 10
G


G
ENTER
YES OR NO
LAST ENTRY?
YES
A
NO
STORE FINAL
RESULTS ON
LU2
RETURN

CRT TERMINAL MESSAGES: SUBROUTINE RSULTS

MESSAGE NO.	FORMAT NO.	MESSAGE
1	20	WERE THESE DECAYS PERFORMED IN THE MODEL REVERBERATION ROOM?
2	22	ENTER NO. OF MICROPHONES USED IN THIS MEASUREMENT (2 DIGITS)
3	30	ENTER VANE SPEED IN REV/MIN (XX.X)
4	50	DESCRIBE SOUND SOURCE (1 LINE)
5	60	DESCRIBE ROOM CONTENTS (1 LINE)
6	80	ENTER DATE
7	100	ENTER TIME OF DAY
8	160	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE
9	900	DO YOU WANT ANOTHER PRINT OF THESE RESULTS?
10	920	DO YOU WANT TO CORRECT ANY TYPING MISTAKES IN THE PRINTOUT?

PRINTOUT 1 - SUBROUTINE RSULTS

PRINTOUT 2 - SUBROUTINE RSULTS

REVERBERATION DECAY MEASUREMENTS

BAND NO.	FREQ. (HZ)	ABSORPTION (METERS*2)	REVERB. TIME (SEC)	STD.DEV. (SEC)	RELATIVE STD.DEV. (%)	CONFIDENCE INTERVAL (%.+OR-)	NO. OF DECAYS
14	25	3.08	22.11	3.66	16.56	7.7	20
15	32	3.83	17.82	4.02	22.57	10.3	21
16	40	4.59	14.85	3.53	23.78	10.0	24
17	50	7.17	9.51	2.55	26.82	11.6	23
18	63	16.46	4.14	0.57	13.68	5.8	24
19	80	14.92	4.57	0.58	12.59	5.3	24
20	100	22.58	3.02	0.45	14.75	6.2	24
21	125	18.38	3.71	0.35	9.45	4.0	24
22	160	15.12	4.51	0.26	5.68	2.4	24
23	200	13.14	5.19	0.21	3.96	1.7	24
24	250	10.80	6.31	0.17	2.72	1.2	24
25	315	10.03	6.80	0.15	2.36	1.0	24
26	400	9.79	6.97	0.17	2.38	1.0	24
27	500	10.09	6.76	0.15	2.21	0.9	24
28	630	11.04	6.18	0.16	2.60	1.1	24
29	800	11.53	5.91	0.11	1.89	0.8	24
30	1000	11.49	5.94	0.08	1.35	0.6	24
31	1250	11.52	5.92	0.07	1.26	0.5	24
32	1600	12.04	5.66	0.07	1.26	0.5	24
33	2000	13.13	5.19	0.09	1.83	0.8	24
34	2500	14.71	4.63	0.07	1.41	0.6	24
35	3150	17.02	4.01	0.06	1.56	0.7	24
36	4000	20.40	3.34	0.05	1.48	0.6	24
37	5000	25.26	2.70	0.04	1.41	0.6	24
38	6300	32.09	2.12	0.05	2.12	0.9	24
39	8000	43.05	1.58	0.02	1.57	0.7	24
40	10000	58.69	1.16	0.03	2.35	1.0	24
41	12500	81.50	0.84	0.03	3.02	1.3	24
42	16000	113.98	0.61	0.04	6.52	2.8	24
43	20000						0

TIME BETWEEN SAMPLES = 0.10017 SECONDS
NO. OF SAMPLES IN SIGNAL MEASUREMENT = 40
NO. OF SAMPLES IN DECAY MEASUREMENT = 160
NO. OF DECAYS CONDUCTED = 24
NO. OF MICROPHONES = 12

CURVE FITTING LIMITS:
5.0 DB DOWN FROM SIGNAL LEVEL
10.0 DB UP FROM NOISE LEVEL

CONFIDENCE LEVEL = 95.00 %

DRY BULB TEMPERATURE = 21.8 DEG. C (71.2 DEG. F)
WET BULB TEMPERATURE = 16.7 DEG. C (62.0 DEG. F)
RELATIVE HUMIDITY = 59.7 %
BAROMETRIC PRESSURE = 980.8 MILLIBARS (735.7 MM HG)

VANE SPEED: 7.5 RPM
SOUND SOURCE: MCINTOSH MODEL ML-1C SPEAKER IN (0 0.1) CORNER: PINK NOISE: 10 VRMS
ROOM CONTENTS: 12 LOW FREQUENCY ABSORBERS

DATE: 3 18 77
TIME: 7:30 A.M.

SUBROUTINE RSULTS

```
$ASSM
RSULTS PROG RSULTS - SUBROUTINE FOR DISPLAYING RESULTS OF MANY DECAYS
$FORT
C    ----------  --------------------
C    OTHER SUBROUTINES CALLED:  HUMID
C    ------------------------------
       SUBROUTINE RSULTS(X,X0,ARRAY1,ARRAY2,FREQ,NDCAYS,CLEVEL,NSIG,
      1NDEC,LINE,DELTA,BLOW,BHIGH,CUPPER,CLOWER)
       INTEGER*2 B,NSIG,NDEC,FREQ(30),LU,FF,BLANK,STAR,PLUS,
      1LINE(121),X0(30),BLOW,BHIGH
       DIMENSION X(60),ARRAY1(30),ARRAY2(30),DATE(5),TIME(5),
      1STATUS(18),SPEAKR(18)
       DATA FF,BLANK,STAR,PLUS/X'0C00',X'2000',X'2A00',X'2B00'/
 10    CALL HUMID(HUMDTY,DTEMPF,DTEMPC,WTEMPF,WTEMPC,PRESS1,PRESS2,
      1VSOUND)
       WRITE (5,20)
 20    FORMAT ('WERE THESE DECAYS PERFORMED IN THE MODEL REVERBERATION
      1 ROOM?')
       READ (5,910) GS
       MODEL=1
       IF (GS.EQ.'NO') MODEL=0
       WRITE (5,22)
 22    FORMAT ('ENTER NO. OF MICROPHONES USED IN THIS MEASUREMENT
      1   (2 DIGITS)')
       READ (5,24) NMIC
 24    FORMAT (I2)
       WRITE (5,30)
 30    FORMAT ('ENTER VANE SPEED IN REV/MIN   (XX.X)')
       READ (5,40) VANE
 40    FORMAT (F10.5)
       WRITE (5,50)
 50    FORMAT ('DESCRIBE SOUND SOURCE   (1 LINE)')
       READ (5,70) (SPEAKR(I),I=1,18)
       WRITE (5,60)
 60    FORMAT (X/'DESCRIBE ROOM CONTENTS (1 LINE)')
       READ (5,70)(STATUS(I),I=1,18)
 70    FORMAT (18A4)
       WRITE (5,80)
 80    FORMAT (X/'ENTER DATE')
       READ (5,90)(DATE(I),I=1,5)
 90    FORMAT (5A4)
       WRITE (5,100)
 100   FORMAT ('ENTER TIME OF DAY')
       READ (5,90) (TIME(I),I=1,5)
       VOLUME=424.8
       IF (MODEL.EQ.1) VOLUME = 6.6375
       DO 140 I=1,30
       IF (X0(I).EQ.0)GO TO 140
       X(I)=.921*60*VOLUME/(ARRAY1(I)*VSOUND)
 140   CONTINUE
 150   WRITE (5,160)
 160   FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE')
       READ (5,170)LU
 170   FORMAT (I1)
       WRITE (LU,300)
 300   FORMAT (X/X/X/51X,'ABSORPTION  (SQUARE METERS)'/1/X/6X,'0',
      19X,'10',8X,'20',8X,'30',8X,'40',8X,'50',8X,'60',8X,'70',8X,
      2'80',8X,'90',7X,'100',7X,'110',7X,'120'/6X,25('.',4X)/X/X)
       DO 350 B=BLOW,BHIGH
       I=B-13
```

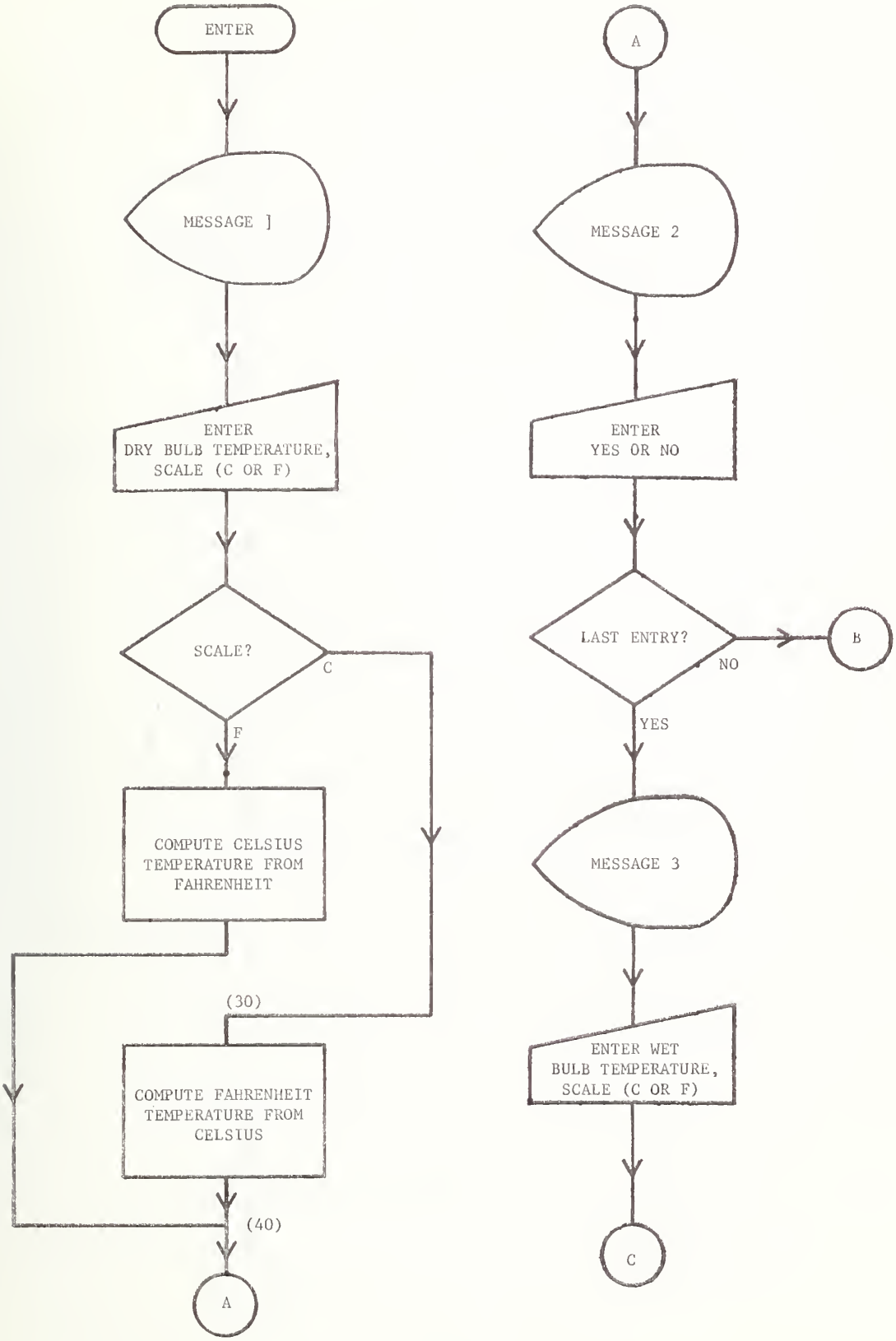
```
      DO 310 J=1,121
      LINE(J)=BLANK
310   CONTINUE
      IF (X0(I).EQ.0) GO TO 330
      J=X(I)+1.5
      IF (J.LT.1) GO TO 330
      IF (J.GT.121) GO TO 320
      LINE(J)=STAR
      GO TO 330
320   LINE(121)=PLUS
330   WRITE(LU,340)FREQ(I),(LINE(J),J=1,121)
340   FORMAT(15,X,121A1)
350   CONTINUE
      WRITE (LU,360)FF
360   FORMAT (X/'FREQUENCY'/2X,'(HZ)'/A1)
      WRITE (LU,500)
500   FORMAT (T21,'REVERBERATION DECAY MEASUREMENTS'/X/X/X/
     1'BAND',T28,'REVERB.',T47,'RELATIVE',T57,'CONFIDENCE',
     2T69,'NO. OF'
     3T2,'NO.',T8,'FREQ.',T15,'ABSORPTION',T29,'TIME',T37,'STD.DEV.',
     4T47,'STD.DEV.',T58,'INTERVAL',T69,'DECAYS'/
     5T8,'(HZ)',T15,'(METERS**2)',T28,'(SEC)',T38,'(SEC)',
     6T49,'(%)',T58,'(%,+OR-)',/)
      DO 600 B=BLOW,BHIGH
      I=B-13
      IF (X0(I).EQ.0)GO TO 550
      IF (X0(I).EQ.1)GO TO 570
      C=100.*ARRAY2(I)/ARRAY1(I)
      WRITE (LU,520)B,FREQ(I),X0(I),ARRAY1(I),ARRAY2(I),C,X(I+30),X0(I)
520   FORMAT (T2,I2,T7,I5,T15,F7.2,T27,F6.2,T37,F6.2,T46,F7.2,
     1T59,F5.1,T69,I4)
      GO TO 600
550   WRITE (LU,560)B,FREQ(I)
560   FORMAT (X,I2,3X,I5,T72,'0')
      GO TO 600
570   WRITE (LU,580)B,FREQ(I),X0(I),ARRAY1(I)
580   FORMAT (T2,I2,T7,I5,T15,F7.2,T27,F6.2,T72,'1')
600   CONTINUE
      WRITE (LU,620) DELTA,NSIG,CUPPER,NDEC,CLOWER,NDAYS,
     1NOISE,CLEVEL,DTEMPC,DTEMPF
620   FORMAT (//,X,'TIME BETWEEN SAMPLES = ',F7.5,' SECONDS',
     1T54,'CURVE FITTING LIMITS:'
     2'NO. OF SAMPLES IN SIGNAL MEASUREMENT = ',I3,
     3T48,F5.1,' DB DOWN FROM SIGNAL LEVEL'/
     4'NO. OF SAMPLES IN DECAY  MEASUREMENT = ',I3,
     5T48,F5.1,' DB  UP  FROM NOISE  LEVEL'/
     6'NO. OF DECAYS CONDUCTED =',I4/
     7'NO. OF MICROPHONES = ',I2,/
     8'CONFIDENCE LEVEL = ',F5.3,' %',/
     9'DRY BULB TEMPERATURE = ',F4.1,' DEG. C       (',F4.1,' DEG. F)')
      IF (WTEMPC.EQ.-1000.) GO TO 640
      WRITE (LU,630) WTEMPC,WTEMPF
630   FORMAT ('WET BULB TEMPERATURE = ',F4.1,' DEG. C
     1       (',F4.1,' DEG. F)')
640   WRITE (LU,650) HUMDTY,PRESS2,PRESS1
650   FORMAT ('RELATIVE HUMIDITY    = ',F4.1,' %'/
     1'BAROMETRIC PRESSURE  = ',F6.1,' MILLIBARS  (',
     2F5.1,' IN HG)',/)
      WRITE (LU,660) VANE,(SPEED(I),I=1,18),(STATUS(I),I=1,18)
660   FORMAT ('VANE SPEED.     ',F4.1,' RPM'/
```

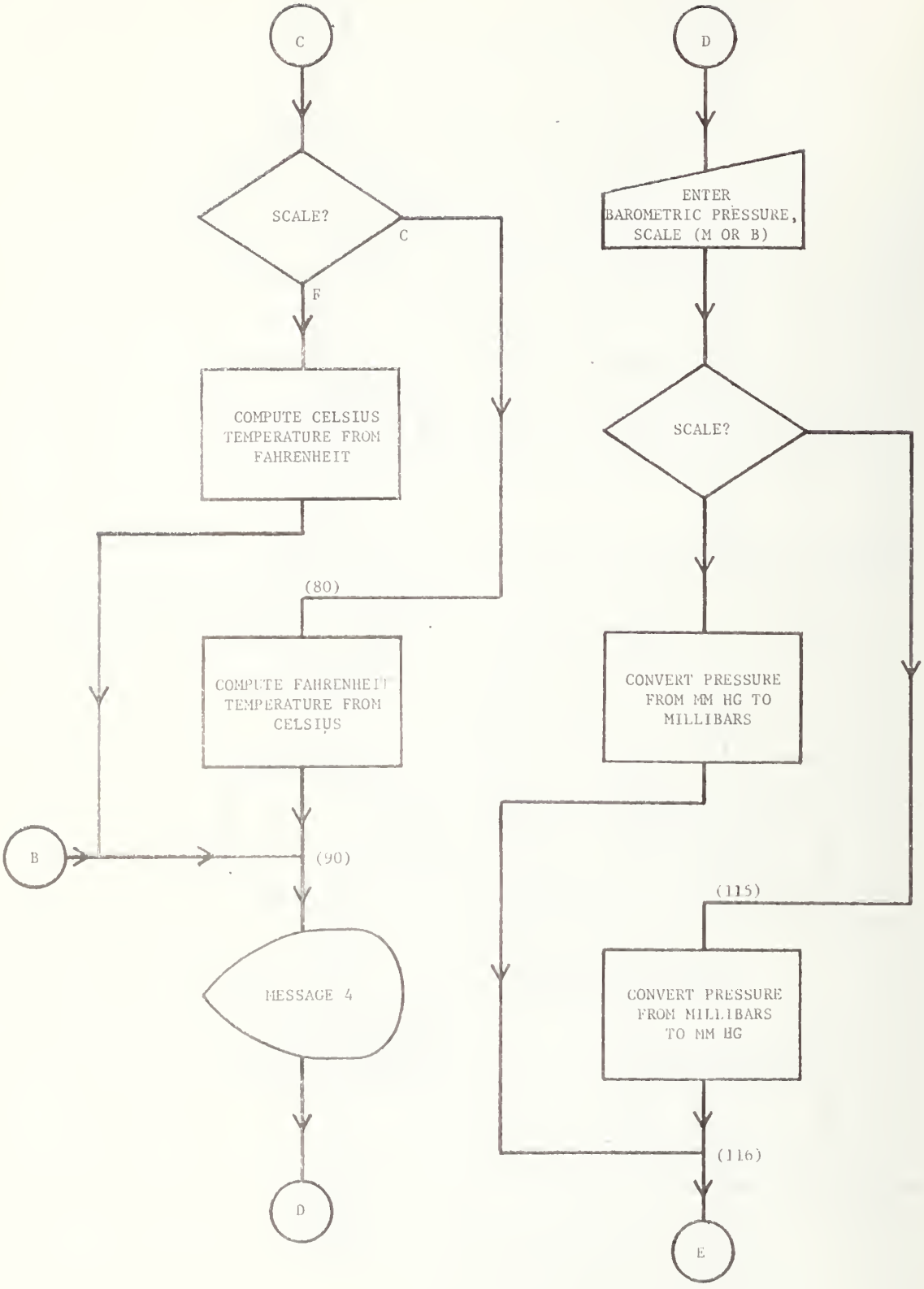
```
121       1'SOUND SOURCE:    '.18A4/'ROOM CONTENTS:  '.18A4)
122        IF (MODEL.EQ.0) GO TO 680
123        WRITE (LU.670)
124 670    FORMAT ('MODEL ROOM')
125 680    WRITE (LU.690) (DATE(I).I=1.5).(TIME(I).I=1.5).FF.FF
126 690    FORMAT (%/'DATE:  '.5A4/'TIME:  '.5A4/A1/A1)
127        WRITE (5.900)
128 900    FORMAT ('DO YOU WANT ANOTHER PRINT OF THESE RESULTS?')
129        READ (5.910)GS
130 910    FORMAT (A4)
131        IF (GS.NE.'NO')GO TO 150
132        WRITE (5.920)
133 920    FORMAT ('DO YOU WANT TO CORRECT ANY TYPING MISTAKES IN THE
134       1 PRINTOUT?')
135        READ (5.910)GS
136        IF (GS.NE.'NO')GO TO 10
137        REWIND 2
138        WRITE (2.925) (DATE(I).I=1.5).(TIME(I).I=1.3)
139 925    FORMAT (5A4.3A4)
140        DO 950 I=1.30
141        B=I+13
142        WRITE (2.930) B.FREQ(I).X(I).ARRAY1(I).ARRAY2(I).X(I+30).X0(I)
143 930    FORMAT (I2.X.I5.X.F8.4.X.F8.4.X.F9.5.X.F6.3.X.I3)
144 950    CONTINUE
145        WRITE (2.960) DTEMPC.HUMDTY.PRESS2
146 960    FORMAT (3F6.1)
147        RETURN
148        END
```

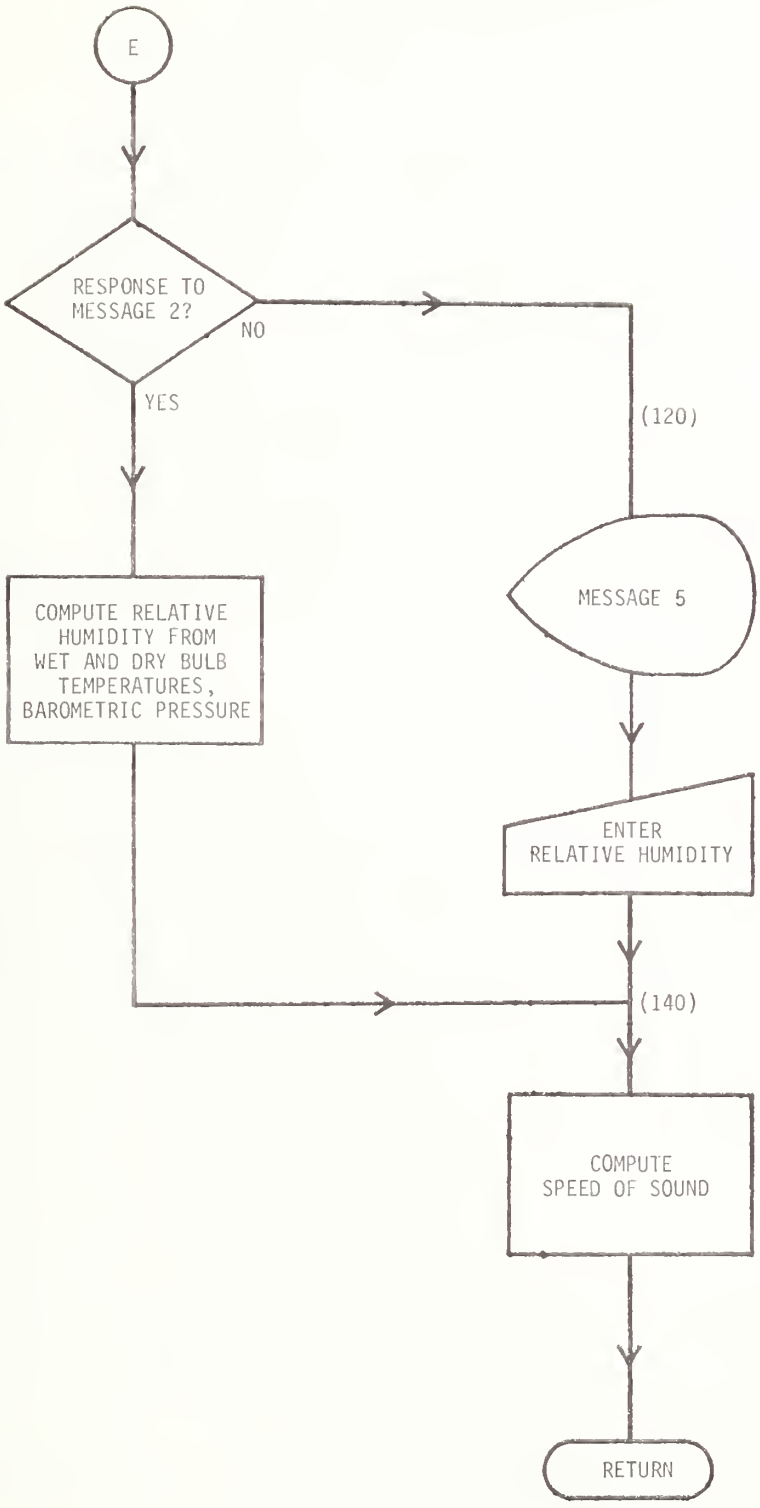
APPENDIX H

Subroutine HUMID Flow Chart, Terminal Messages, and Listings

FLOW CHART: SUBROUTINE HUMID




C
SCALE?
C
F
COMPUTE CELSIUS TEMPERATURE FROM FAHRENHEIT
(80)
COMPUTE FAHRENHEIT TEMPERATURE FROM CELSIUS
B
(90)
MESSAGE 4
D
D
ENTER BAROMETRIC PRESSURE, SCALE (M OR B)
SCALE?
CONVERT PRESSURE FROM MM HG TO MILLIBARS
(115)
CONVERT PRESSURE FROM MILLIBARS TO MM HG
(116)
E


E
RESPONSE TO MESSAGE 2?
NO
YES
(120)
COMPUTE RELATIVE HUMIDITY FROM WET AND DRY BULB TEMPERATURES, BAROMETRIC PRESSURE
MESSAGE 5
ENTER RELATIVE HUMIDITY
(140)
COMPUTE SPEED OF SOUND
RETURN

CRT TERMINAL MESSAGES: SUBROUTINE HUMID

MESSAGE NO.	FORMAT NO.	MESSAGE
1	10	ENTER DRY BULB TEMPERATURE (C OR FXX.X)
2	50	DO YOU WANT TO COMPUTE THE HUMIDITY FROM THE WET BULB TEMPERATURE?
3	70	ENTER WET BULB TEMPERATURE (C OR FXX.X)
4	100	ENTER BAROMETRIC PRESSURE (CHOOSE ONE OF THE FOLLOWING) MXXX.X (IN MILLIMETERS OF HG) BXXXX.X (IN MILLIBARS)
5	130	ENTER THE RELATIVE HUMIDITY IN PERCENT (XX.X)

SUBROUTINE HUMID

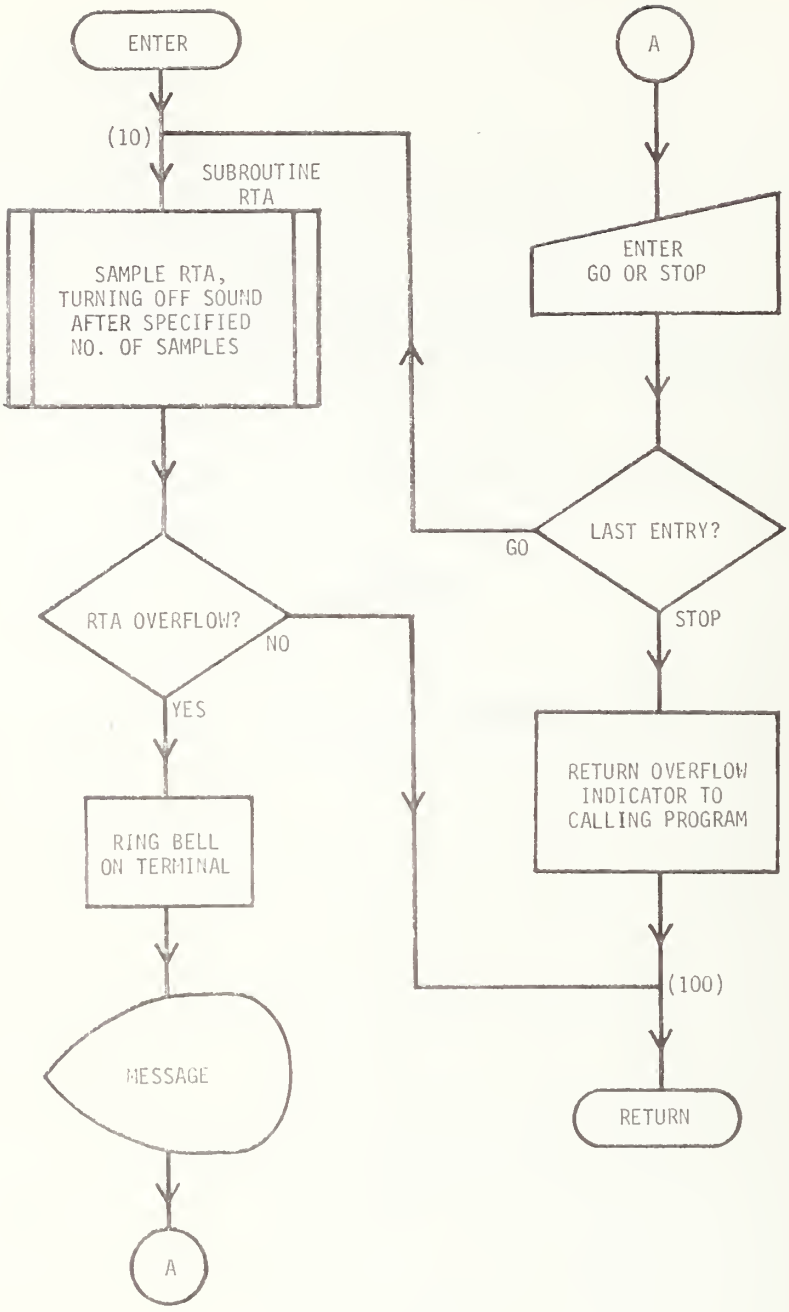
```
$ASSN
HUMID PROG HUMID - SUBROUTINE FOR COMPUTING HUMIDITY FROM WET BULB TEMP
$FORT
C    ------------------------------
C    THIS SUBROUTINE ALLOWS THE ENTERING OF ATMOSPHERIC FACTORS OF
C    TEMPERATURE, RELATIVE HUMIDITY AND PRESSURE.  HUMIDITY CAN BE
C    COMPUTED FROM DRY AND WET BULB TEMPERATURE READINGS.
C    THE SPEED OF SOUND IS ALSO COMPUTED.
C
C    FORTRAN CALL STATEMENT:
C   -CALL HUMID (HUMDTY,DTEMPF,DTEMPC,WTEMPF,WTEMPC,
C                PRESS1,PRESS2,VSOUND)
C
C    HUMDTY (REAL) = RELATIVE HUMIDITY      (PERCENT)
C    DTEMPF (REAL) = DRY BULB TEMPERATURE   (FAHRENHEIT)
C    DTEMPC (REAL) = DRY BULB TEMPERATURE   (CELSIUS)
C    WTEMPF (REAL) = WET BULB TEMPERATURE   (FAHRENHEIT)
C    WTEMPC (REAL) = WET BULB TEMPERATURE   (CELSIUS)
C                    NOTE:  WTEMPC IS SET = -1000. IF YOU CHOOSE
C                           NOT TO ENTER THE WET BULB TEMPERATURE.
C    PRESS1 (REAL) = BAROMETRIC PRESSURE    (MM HG)
C    PRESS2 (REAL) = BAROMETRIC PRESSURE    (MILLIBARS)
C    VSOUND (REAL) = SPEED OF SOUND         (METERS/SECOND)
C
C    NO OTHER SUBROUTINES CALLED
C    ------------------------------
      SUBROUTINE HUMID(HUMDTY,DTEMPF,DTEMPC,WTEMPF,WTEMPC,
     1PRESS1,PRESS2,VSOUND)
      WRITE (5,10)
10    FORMAT ('ENTER DRY BULB TEMPERATURE    (C OR F)XX.X)')
      READ (5,20)SCALE,TEMP
20    FORMAT (A1,F4.1)
      IF (SCALE.EQ.'C') GO TO 30
      DTEMPF=TEMP
      DTEMPC=5*(TEMP-32)/9
      GO TO 40
30    DTEMPF=32+9*TEMP/5
      DTEMPC=TEMP
40    WRITE (5,50)
50    FORMAT ('DO YOU WANT TO COMPUTE THE HUMIDITY FROM THE WET BULB
     1 TEMPERATURE?')
      READ (5,60) GS
60    FORMAT (A4)
      IF (GS.EQ.'NO') GO TO 90
      WRITE (5,70)
70    FORMAT ('ENTER WET BULB TEMPERATURE    (C OR F)XX.X)')
      READ (5,20)SCALE,TEMP
      IF (SCALE.EQ.'C') GO TO 80
      WTEMPF=TEMP
      WTEMPC=5*(TEMP-32) 9
      GO TO 90
80    WTEMPF=32+9*TEMP/5
      WTEMPC=TEMP
90    WRITE (5,100)
100   FORMAT ('ENTER BAROMETRIC PRESSURE  (CHOOSE ONE OF
     1 THE FOLLOWING)',/
     2'      M)XX.X    (IN MILLIMETERS OF HG)',/
     3'      B)XXX.X   (IN MILLIBARS)')
      READ (5,110) SCALE,PRESS
110   FORMAT (A1,F6.1)
```

```
      IF (SCALE.EQ.'B') GO TO 115
      PRESS1=PRESS
      PRESS2=PRESS1*1.33322
      GO TO 116
115   PRESS2=PRESS
      PRESS1=PRESS2*.750064
116   IF (GS.EQ.'NO') GO TO 120
      TEMP=DTEMPC+273.16
      ED=23.5315-(2939/TEMP)-4.922*ALOG10(TEMP)
      ED=(1013.25/1333)*10.**ED
      TEMP=WTEMPC+273.16
      EW=23.5315-(2939/TEMP)-4.922*ALOG10(TEMP)
      EW=(1013.25/1333)*10.**EW
      E=EW-.00066*PRESS1*(DTEMPC-WTEMPC)*(1+.00115*WTEMPC)
      HUMDTY=100*E/ED
      GO TO 140
120   WTEMPC=-1000.
      WRITE (5,130)
130   FORMAT ('ENTER THE RELATIVE HUMIDITY IN PERCENT   (XX.X)')
      READ (5,135) HUMDTY
135   FORMAT (F5.2)
140   vSOUND=331.43+.59*DTEMPC
      RETURN
      END
```

APPENDIX I

Subroutine SAMPLE Flow Chart and Listings

FLOW CHART: SUBROUTINE SAMPLE



MESSAGE: YOU HAVE SIGNAL LEVEL OVERFLOW ON BAND XX, POINT XXX, LEVEL = XXX.XX
ENTER "GO" TO REPEAT MEASUREMENT, OR "STOP" TO INTERRUPT PROGRAM

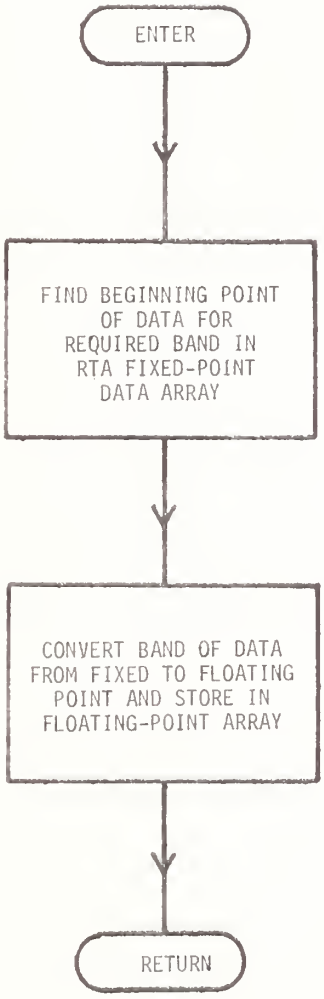
SUBROUTINE SAMPLE

```
$ASSM
SAMPLE PROG SAMPLE - SUBROUTINE FOR TAKING RTA SAMPLES
$FORT
C     ------------------------------
C     THIS SUBROUTINE TAKES A SPECIFIED NUMBER OF RTA SAMPLES AND
C     CHECKS FOR OVERFLOW.
C     FORTRAN CALL STATEMENT:
C    -CALL SAMPLE (NSOUND,NSAMP,TX100,OVFLOW,X0,NX0)
C     NSOUND (INT*2) = NO. OF SAMPLES TO BE TAKEN BEFORE SOUND CUT-OFF
C     NSAMP  (INT*2) = TOTAL NO. OF SAMPLES TO BE TAKEN  (MAX = 200)
C     TX100  (INT*2) = INTEGRATION TIME IN SECONDS X 100
C     OVFLOW (INT*2) = 1 IF AN OVERFLOW CONDITION EXISTS, = 0 OTHERWISE;
C                      (THIS IS AN OPTIONAL PARAMETER FOR THE FORTRAN
C                      CALLING PROGRAM)
C     X0     (INT*2) = ARRAY CONTAINING THE FIXED-POINT RTA DATA
C     NX0    (INT)   = NO. OF ELEMENTS IN THE ARRAY X0:
C                      SHOULD BE CHOSEN SO THAT NX0=(30*NSAMP)+60
C
C     OTHER SUBROUTINES CALLED:  RTA
C     ------------------------------
      SUBROUTINE SAMPLE (NSOUND,NSAMP,TX100,OVFLOW,X0,NX0)
      INTEGER*2 NSOUND,NSAMP,TX100,OVFLOW,X0(NX0),BELL,I
      DATA BELL/X'0700'/
      OVFLOW=0
10    CALL RTA(NSOUND,NSAMP,TX100,X0)
      IF (X0(1).NE.0) GO TO 100
      I=X0(3)+1
      X=X0(4)/100.
      WRITE (5,20) BELL
20    FORMAT (A1)
      WRITE (5,30) X0(2),I,X
30    FORMAT ('YOU HAVE SIGNAL LEVEL OVERFLOW ON BAND ',I2,
     1', POINT ',I3,' , LEVEL = ',F8.2/
     2'ENTER "GO" TO REPEAT MEASUREMENT, OR "STOP" TO INTERRUPT
     3 PROGRAM')
      READ (5,40) GS
40    FORMAT (A4)
      IF (GS.NE.'STOP') GO TO 10
      OVFLOW=1
100   RETURN
      END
```

APPENDIX J

Subroutine BAND Flow Chart and Listings

FLOW CHART: SUBROUTINE BAND



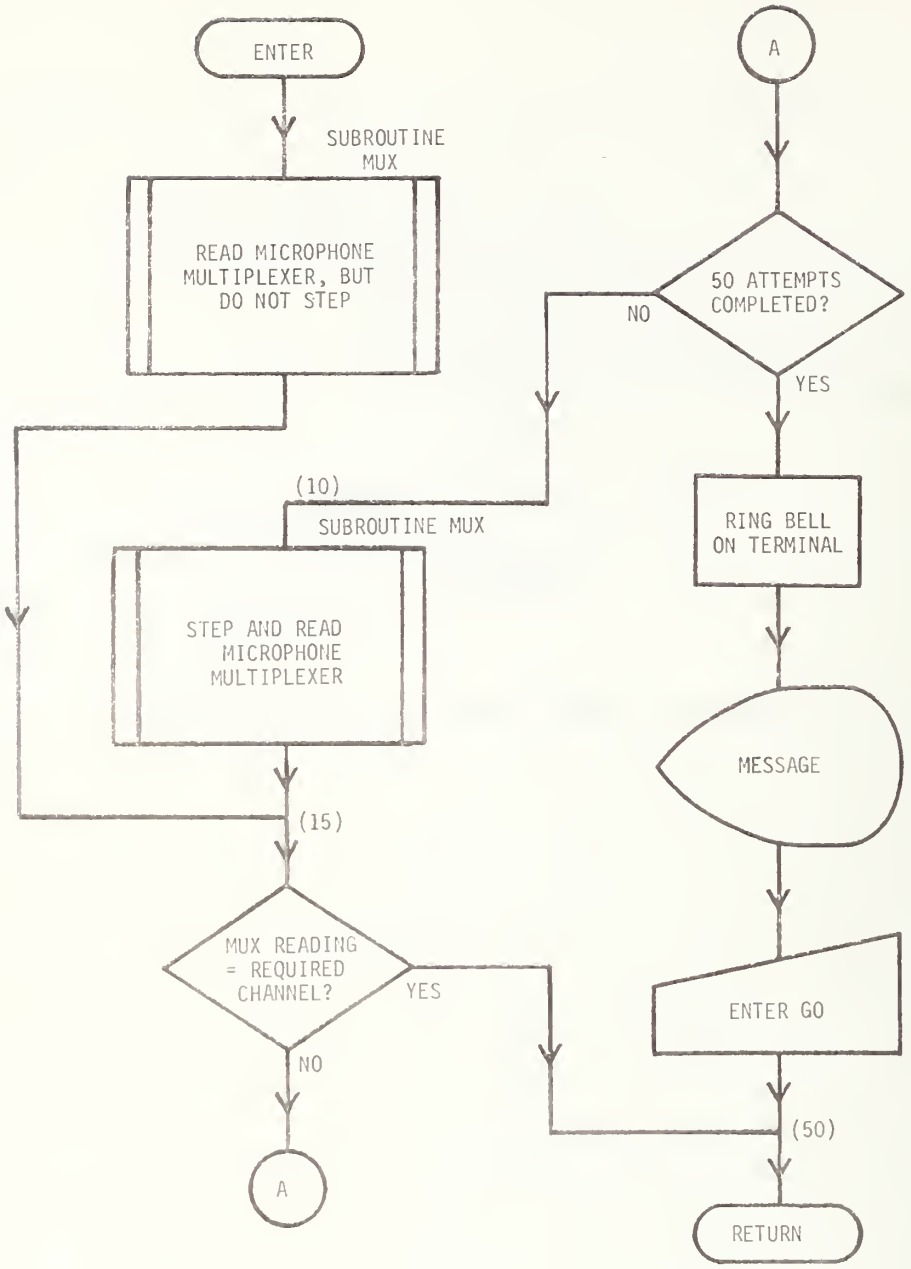
SUBROUTINE BAND

```
$ASSM
BAND PROG BAND - SUBROUTINE FOR CONVERTING ONE BAND OF RTA SAMPLES
$FORT
C     ------------------------------
C     THIS SUBROUTINE CONVERTS ONE BAND OF RTA SAMPLES TO FLOATING POINT
C     FORTRAN CALL STATEMENT:
C    -CALL BAND (BANDNO,NSAMP,X,X0,NX0)
C     BANDNO (INT*2) = BAND NUMBER  (BAND 14 = 25 HZ)
C     NSAMP  (INT)   = NO. OF SAMPLES TAKEN BY RTA  (MAX = 200)
C     X      (REAL)  = ARRAY TO CONTAIN THE FLOATING POINT RTA DATA
C                      FOR THE SPECIFIED BAND
C     X0     (INT*2) = ARRAY CONTAINING THE FIXED-POINT RTA DATA
C     NX0    (INT)   = NO. OF ELEMENTS IN THE ARRAY X0:
C                      SHOULD BE CHOSEN SO THAT NX0=(30*NSAMP)+60
C
C     NO OTHER SUBROUTINES CALLED
C     ---------------------------
      SUBROUTINE BAND(BANDNO,NSAMP,X,X0,NX0)
      DIMENSION X(NSAMP)
      INTEGER*2 BANDNO,X0(NX0),I,INDEX,ISTPT
      ISTPT=(BANDNO-14)*(NSAMP+2)+2
      DO 10 I=1,NSAMP
      INDEX=ISTPT+I
      X(I)=X0(INDEX)/100.
10    CONTINUE
      RETURN
      END
```

APPENDIX K

Subroutine SUBMUX Flow Chart and Listings

FLOW CHART: SUBROUTINE SUBMUX



MESSAGE: THE PROPER MICROPHONE CANNOT BE SELECTED AUTOMATICALLY
MANUALLY STEP MULTIPLEXER TO MICROPHONE XX
ENTER "GO" TO CONTINUE

SUBROUTINE SUBMUX

```
$ASSM
SUBMUX PROG SUBMUX - SUBROUTINE TO SET MULTIPLEXER TO A GIVEN MICROPHONE
$FORT
C    ------------------------------
C    THIS SUBROUTINE STEPS THE MICROPHONE MULTIPLEXER TO A GIVEN MICROPHONE
C    CHANNEL.
C
C    FORTRAN CALL STATEMENT:
C   -CALL SUBMUX(J)
C    J    (INTEGER)   =   THE MICROPHONE CHANNEL TO BE SELECTED
C
C    OTHER SUBPROGRAMS CALLED:  MUX
C    ------------------------------
       SUBROUTINE SUBMUX(J)
       INTEGER*2 BELL
       DATA BELL/X'0700'/
       I=0
       CALL RDMUX(MXRDG)
       GO TO 15
 10    CALL MUX(MXRDG)
 15    IF (MXRDG.EQ.J) GO TO 50
       I=I+1
       IF (I.LT.50) GO TO 10
       WRITE (5,20) BELL
 20    FORMAT (A1)
       WRITE (5,30) J
 30    FORMAT (X/'THE PROPER MICROPHONE CANNOT BE SELECTED
      1 AUTOMATICALLY'/'MANUALLY STEP MULTIPLEXER TO MICROPHONE ',I2/
      2'ENTER "GO" TO CONTINUE')
       READ (5,40) OS
 40    FORMAT (A4)
 50    RETURN
       END
```

APPENDIX L

Listings for Function Subprograms TINORM and STUDIM

FUNCTION SUBPROGRAM TINORM

```
$ASSM
TINORM PROG TINORM - SUBPROG. GIVING INVERSE OF NORMAL DISTRIBUTION
$FORT
C    ------------------------------
C    THIS SUBPROGRAM COMPUTES THE VALUE OF THE INVERSE OF THE STANDARD
C    NORMAL DISTRIBUTION FUNCTION.  IT IS A VERSION OF THE UNIVAC 1108
C    STAT-PACK POUTINE TINORM.
C
C    FORTRAN CALLING SEQUENCE:
C  > VARIABLE = TINORM(ALPHA)
C
C    TINORM (REAL) = THE CALCULATED VALUE OF THE INVERSE OF THE NORMAL
C                    DISTRIBUTION: THIS IS THE ORDINATE. OR NUMBER OF
C                    STANDARD DEVIATIONS FROM THE MEAN CORRESPONDING TO
C                    THE GIVEN VALUE OF ALPHA.
C    ALPHA  (REAL) = THE VALUE OF THE PROBABILITY FOR WHICH THE ORDINATE
C                    IS TO BE CALCULATED: THIS IS EQUAL TO 1.0 MINUS
C                    THE AREA UNDER THE CURVE FROM TINORM TO INFINITY.
C                    (ALPHA MUST LIE BETWEEN 0.0 AND 1.0)
C
C    NO OTHER SUBPROGRAMS CALLED
C    ------------------------------
      FUNCTION TINORM(ALPHA)
      DIMENSION A(3),B(3)
      DATA (A(I),I=1,3)/.010328,.802853,2.515517/,
     1(B(I),I=1,3)/.001308,.189269,1.432788/
      X=ALPHA
      IF (X.GT..5) X=1.-X
      X=SQRT(-2.*ALOG(X))
      TINORM=X-(A(3)+X*(A(2)+X*A(1)))/(1.+X*(B(3)+X*(B(2)+X*B(1))))
      IF (ALPHA.LT..5) TINORM=-TINORM
      RETURN
      END
```

FUNCTION SUBPROGRAM STUDIM

```
$ASSM
STUDIM PROG STUDIM - SUBPROG. SUBSTITUTING FOR STUDIN
$FORT
C    ------------------------------
C    THIS SUBPROGRAM IS A SUBSTITUTE FOR STUDIN. GIVING THE VALUE OF
C    THE INVERSE OF STUDENT'S T-DISTRIBUTION BY READING FROM A TABLE
C    GENERATED ON A STORAGE FILE BY THE PROGRAM GEN1.
C
C    IT IS INTENDED FOR USE IN THOSE CASES WHEN THE MEMORY SPACE
C    REQUIRED BY STUDIN IS NOT AVAILABLE.
C
C    THE FILE MUST BE GENERATED FOR THE INTENDED VALUE OF ALPHA:
C    ALPHA HERE HAS THE SAME MEANING AS IN THE PROGRAM STUDIN. AND
C    THE VALUE STUDIM CORRESPONDS TO THE VALUE STUDIN.
C
C    THE FILE MUST BE A RANDOM RECORD ACCESS FILE.
C
C    FORTRAN CALLING SEQUENCE:
C  > VARIABLE = STUDIM(ALPHA,N,LUFILE,LUERR)
C
C    STUDIM (REAL) = THE CALCULATED VALUE OF T
C    ALPHA  (REAL) = THE CONFIDENCE COEFFICIENT
C    LUFILE (INT)  = LOGICAL UNIT OF THE STORAGE FILE
C    LUERR  (INT)  = LOGICAL UNIT FOR ERROR MESSAGE PRINTOUT
C                    (USUALLY LUERR = 5)
C    ------------------------------
      FUNCTION STUDIM(ALPHA,N,LUFILE,LUERR)
      DIMENSION T(50)
      INTEGER*2 BELL
      DATA BELL/X'0700'/
10    READ (LUFILE,REC=0) ALPHA1
      IF (ABS(ALPHA-ALPHA1).LE..0001) GO TO 30
      WRITE (LUERR,20) BELL,LUFILE,ALPHA1,ALPHA,LUFILE
20    FORMAT (A1,/,/,'THE STORAGE FILE ON LU',I2,' CONTAINS THE TABLE
     1 FOR ALPHA = ',F6.4
     2'THE REQUIRED T-DISTRIBUTION TABLE IS FOR ALPHA = ',F6.4,/,/
     3'REASSIGN LU',I2,' TO THE PROPER FILE',/
     4'ENTER "CONTINUE" WHEN READY')
      PAUSE1
      GO TO 10
30    N1=N
      IF (N1.GT.1000) N1=1000
      NSECTR=1+(N1-1)/50
      READ (LUFILE,REC=NSECTR) T
      J=N1-50*(NSECTR-1)
      STUDIM=T(J)
      RETURN
      END
```

APPENDIX M

Listings for Interdata Assembly Language Subroutines

MUX, RTA, SOUND, and DELAY

SUBROUTINE MUX

```
 1 MUX      PROG  MUX                      - PROGRAM FOR STEPPING AND READING MUX
 2          WIDTH  132
 3 *    --------------------
 4 *    FORTRAN CALL STATEMENTS:
 5 *   -CALL MUX(X) --- STEP AND READ
 6 *   -CALL RDMUX(X) --- READ ONLY
 7 *    PARAMETER = MULTIPLEXER READING
 8 *    --------------------
 9          ENTRY  MUX
10          ENTRY  RDMUX
11 UIM      EQU   12
12 WR       EQU   13
13 INDEX    EQU   14
14 R15      EQU   15
15 MUX      STM   12,RETURN              START HERE TO STEP MULTIPLEXER
16          LHI   UIM,X'8B'
17          LIS   WR,10
18          OCR   UIM,WR
19          LIS   WR,1
20          WHR   UIM,WR
21          XHR   WR,WR
22          WHR   UIM,WR
23          B     CONT
24 RDMUX    STM   12,RETURN              START HERE TO READ MULTIPLEXER
25          LHI   UIM,X'8B'
26 CONT     LHI   INDEX,1000             THESE NEXT 4 LINES PROVIDE A DELAY
27 DELAY    SLLS  WP,15
28          SIS   INDEX,1
29          BP    DELAY
30          LIS   WP,11
31          OCR   UIM,WR
32          WHR   UIM,WP
33          PHR   UIM,WP
34          SLLS  WP,12
35          SRLS  WP,12
36          LH    INDEX,2(R15)           FIND FORTRAN ADDRESS OF PARAMETER
37          STH   WP,0(INDEX)            STORE PARAMETER AT ITS FORTRAN ADDRESS
38          LM    12,RETURN
39          AIS   R15,4
40          BR    R15
41 RETURN   DS    8
42          END
```

SUBROUTINE RTA

```
 1 RTA      PROG  RTA                       - PROGRAM FOR DRIVING RTA
 2          WIDTH  132
 3 *    ----------------------------
 4 *    FORTRAN CALL STATEMENT:
 5 *   -CALL RTA(NSOUND.NSAMP.TX100.X0)
 6 *    NSOUND = NO. OF SAMPLES TO BE TAKEN BEFORE SOUND CUT-OFF
 7 *    NSAMP  = TOTAL NO. OF SAMPLES TO BE TAKEN
 8 *    TX100  = INTEGRATION TIME X 100
 9 *    X0     = ARRAY OF 2-BYTE INTEGERS TO CONTAIN THE
10 *             FIXED-POINT RTA DATA
11 *    ----------------------------
12          ENTRY  RTA
13 UIM     EQU   0
14 N1000   EQU   1                       CONTAINS THE INTEGER 1000
15 N100    EQU   2                       CONTAINS THE INTEGER 100
16 N25     EQU   3                       CONTAINS THE INTEGER 25
17 WR      EQU   4
18 WR1     EQU   5
19 WR2     EQU   6
20 ACC     EQU   7
21 MR1     EQU   8                       MULTIPLICATION REGISTERS
22 MR2     EQU   9
23 LR1     EQU   10                      LOOP REGISTERS FOR BXLE LOOP
24 LR2     EQU   11
25 LR3     EQU   12
26 INDEX   EQU   13                      NUMBER OF SAMPLES ALREADY TAKEN
27 R14     EQU   14
28 R15     EQU   15
29 RTA     STM   0.RETURN               ENTRY POINT
30         LHI   UIM.X'8B'
31         LH    WR1.6(R15)             FIND FORTRAN ADDRESS OF TX100
32         LH    WR.0(WR1)              LOAD TX100 INTO WR
33         CHI   WR.10                  DETERMINE INTEGRATION TIME
34         BE    T10
35         CHI   WR.25
36         BE    T25
37         CHI   WR.50
38         BE    T50
39         CHI   WR.100
40         BE    T100
41         CHI   WR.200
42         BE    T200
43         CHI   WR.400
44         BE    T400
45         CHI   WR.800
46         BE    T800
47         CHI   WR.1600
48         BE    T1600
49         CHI   WR.3200
50         BE    T3200
51 T10     LIS   WR.8                   DETERMINE TIME SECTION OF
52         B     TIME                   RTA CONROL WORD
53 T25     LIS   WR.2
54         B     TIME
55 T50     LIS   WR.4
56         B     TIME
57 T100    LIS   WR.6
58         B     TIME
59 T200    LIS   WR.1
60         B     TIME
```

```
 61 T400     LIS    WR.3
 62          B      TIME
 63 T800     LIS    WR.5
 64          B      TIME
 65 T1600    LIS    WR.7
 66          B      TIME
 67 T3200    LIS    WR.9
 68 TIME     SLLS   WR.12
 69          AHI    WR.X'08FF'
 70          STH    WR.RTACWD          SET RTA CONTROL WORD
 71          LH     ACC.8(R15)         LOAD FORTRAN ADDRESS OF X0 INTO ACC
 72          STH    ACC.STORE          STORE ADDRESS OF X0 IN STORE
 73          LH     WR.2(R15)          FIND FORTRAN ADDRESS OF NSOUND
 74          LH     WR1.0(WR)          LOAD NSOUND INTO WR1
 75          STH    WR1.NSOUND         STORE NSOUND IN NSOUND
 76          LH     WP.4(R15)          FIND FORTRAN ADDRESS ON NSAMP
 77          LH     WR1.0(WR)          LOAD NSAMP INTO WR1
 78          STH    WR1.NSAMP          STORE NSAMP IN NSAMP
 79          LHR    WR.WR1
 80          SLLS   WR1.8
 81          AIS    WR.2
 82          LHR    WR2.WR
 83          AHR    WR.WP2             WR NOW CONTAINS 2(NSAMP+2), WHICH IS
 84          XHR    WR2.WP2            THE TOTAL AMOUNT OF STORAGE OCCUPIED
 85          LIS    LR1.1              BY EACH LIST: USED TO UPDATE ACC
 86          LIS    LP2.1
 87          LHI    LP3.30
 88 LOOP     STH    WP1.0(ACC)         STORE LIST PARAMETER BLOCK AT
 89          STH    WP2.2(ACC)         BEGINNING OF 30 CONSECUTIVE LISTS
 90          AHP    ACC.WP             (THESE ARE CIRCULAR LISTS)
 91          BXLE   LP1.LOOP
 92          LHI    N1000.1000
 93          LHI    N100.100
 94          LHI    N25.25
 95          XHR    INDEX.INDEX        SET INDEX=0 (NO. SAMPLES ALREADY TAKEN)
 96 LOOP3    CH     INDEX.NSOUND       IF INDEX = NSOUND, THEN CONTINUE:
 97          BTC    X'3',BRNCH1        OTHERWISE, GO TO BRNCH1
 98          LIS    WR.1
 99          OCR    UIM.WP             SELECT RTA CONTROL LINE
100          LHI    WP.X'10FF'
101          WHR    UIM.WR             STOP RTA
102          XHR    WP.WP
103          OCP    UIM.WP             SELECT SOUND CONTROL LINE
104          XHR    WP.WP
105          WHR    UIM.WR             STOP SOUND
106          BFFS   0,5                BRANCH FORWARD 5 LOCATIONS
107 BRNCH1   CHI    INDEX,0            IF INDEX = 0, THEN CONTINUE:
108          BTC    X'3',BRNCH2        OTHERWISE, GO TO BRNCH2
109          LIS    WR.1
110          OCR    UIM.WP             SELECT RTA CONTROL LINE
111          WH     UIM.RTACWD         SEND RTA CONTROL WORD (SET UP RTA)
112          LIS    WR.4
113          OCR    UIM.WP             SELECT START MEASUREMENT LINE
114          WHR    UIM.WP             START MEASUREMENT
115 BRNCH2   LIS    LR1.1
116          LIS    LR3.6
117 LOOP1    XHP    WR.WP              THROW OUT 1ST 6 READINGS
118          SSR    UIM.WP             (BANDS 12 AND 13)
119          BTBS   X'4',1
120          RHR    UIM.WP
```

```
121           BXLE   LR1.LOOP1
122           LH     ACC.STORE
123           LH     WR2.NSAMP
124           AIS    WR2.2
125           LHR    WR.WR2
126           AHR    WR2.WR                  WR2 NOW CONTAINS 2(NSAMP+2),
127           LIS    LR1.1                   WHICH IS USED TO UPDATE ACC
128           LHI    LR3.30
129 LOOP2     SSR    UIM.WR                  TEST FOR RTA BUSY
130           BTBS   X'4',.1                 WAIT IF BUSY
131           RHR    UIM.WP                  1ST READING (BAND NUMBER) THROWN OUT
132           SSR    UIM.WR
133           BTBS   X'4',.1
134           RHR    UIM.WR                  2ND READING
135           LHR    MR2.WR                  CONVERT 10'S DIGIT
136           SRLS   MR2.12
137           MHR    MR1.N1000               WR1 USED TO ACCUMULATE RESULT OF
138           LHR    WR1.MR2                 BCD TO BINARY CONVERSION
139           LHR    MR2.WR                  CONVERT 1'S DIGIT
140           SLLS   MR2.4
141           SRLS   MR2.12
142           MHR    MR1.N100
143           AHR    WR1.MR2
144           SSR    UIM.WR
145           BTBS   X'4',.1
146           RHR    UIM.WR                  3RD READING
147           LHR    MR2.WR                  CONVERT 1/4 DB DIGIT
148           SLLS   MR2.4
149           SRLS   MR2.12
150           MHR    MR1.N25
151           AHR    WR1.MR2
152           SRLS   WR.12                   CHECK 100'S DIGIT
153           LIS    MR1.1                   (BIT 1 = VALUE OF 100'S DIGIT)
154           CHR    WR.MR1                  (BIT 2 = RTA OVERFLOW BIT)
155           BTC    X'1',.ABL               IF 100'S DIGIT < 1, BRANCH TO ABL
156           BTC    X'2',.OVFLOW            IF 100'S DIGIT > 1, BRANCH TO OVFLOW
157           AHI    WR1.10000               IF 100'S DIGIT = 1, ADD 10000 TO WR1
158           B      ABL
159 OVFLOW    SIS    WR.2                    IN CASE OF OVERFLOW, REMOVE BIT 2
160           LHI    MR2.10000               FROM VALUE OF 100'S DIGIT
161           MHR    MR1.WR
162           AHR    WR1.MR2                 WR1 NOW CONTAINS VALUE OF OVERFLOW
163           LH     ACC.STORE               DATA POINT
164           XHR    WR.WR
165           STH    WR.0(ACC)               STORE "0" AT ADDRESS OF X0(1)
166           AIS    LR1.13
167           STH    LR1.2(ACC)              STORE BAND NUMBER AT X0(2)
168           STH    INDEX.4(ACC)            STORE SAMPLE NUMBER AT X0(3)
169           STH    WR1.6(ACC)              STORE VALUE OF OVERFLOW POINT AT X0(4)
170           B      RTRN                    STOP SAMPLING AND RETURN TO FORTRAN
171 ABL       ABL    WR1.0(ACC)              ADD FINAL NUMBER TO PROPER LIST
172           AHR    ACC.WR2
173           BXLE   LR1.LOOP2
174           AIS    INDEX.1
175           CH     INDEX.NSAMP
176           BTC    X'8',.LOOP3             IF INDEX < NSAMP, RETURN TO LOOP3:
177 RTRN      LM     0.RETURN                OTHERWISE, RETURN TO FORTRAN
178           AIS    R15.10
179           BR     R15
180 RETURN    DS     32
```

```
181 STORE   DS    2                    ADDRESS OF X0(1) (START OF THE X0'S)
182 NSOUND  DS    2
183 NSAMP   DS    2
184 RTACWD  DS    2                    RTA CONTROL WORD
185 *   ------------------------------
186 *   LOOP:   BXLE LOOP: STORES LIST PARAMETER BLOCK
187 *           AT BEGINNING OF 30 CONSECUTIVE CIRCULAR LISTS
188 *   LOOP1:  THROWS OUT 1ST 6 READINGS (BANDS 12 & 13)
189 *   LOOP2:  BXLE LOOP: READS BANDS 14 TO 43. CONVERTS
190 *           DATA FROM BCD TO BINARY. AND STORES THE
191 *           FIXED-POINT DATA IN 30 CONSECUTIVE LISTS
192 *   LOOP3:  ENCLOSES BOTH LOOP1 AND LOOP2:
193 *           REPEATS MEASUREMENT UNTIL NSAMP SAMPLES
194 *           HAVE BEEN TAKEN
195 *   ------------------------------
196         END
```

SUBROUTINE SOUND

```
 1 SOUND   PROG  SOUND                    - PROGRAM FOR TURNING ON SOUND
 2         WIDTH  132
 3 *   --------------------
 4 *   FORTRAN CALL STATEMENT:
 5 *  -CALL SOUND(X)
 6 *   X = SOUND CONTROL WORD
 7 *       SOUND ON:  X = 1
 8 *       SOUND OFF: X = 0
 9 *   --------------------
10         ENTRY  SOUND
11 UIM     EQU   13
12 WR      EQU   14
13 R15     EQU   15
14 SOUND   STM   13,RETURN            ENTRY POINT
15         LHI   UIM,X'8B'
16         XHR   WR,WR
17         OCR   UIM,WR               SELECT SOUND CONTROL LINE
18         LH    WR,2(R15)            FIND FORTRAN ADDRESS OF X
19         LH    WR,0(WR)             LOAD X INTO WR
20         SLLS  WR,8
21         WHR   UIM,WR               SEND SOUND CONTROL WORD
22         LM    13,RETURN
23         AIS   R15,4
24         BR    R15                  RETURN TO FORTRAN
25 RETURN  DS    6
26         END
```

SUBROUTINE DELAY

```
 1 DELAY   PROG  DELAY                  - APPROXIMATELY 1 MILLISECOND INCREMENT
 2         WIDTH  132
 3 *   --------------------
 4 *   FORTRAN CALL STATEMENT:
 5 *  -CALL DELAY(X)
 6 *   X = NUMBER OF 1 MILLISECOND INCREMENTS
 7 *   --------------------
 8         ENTRY  DELAY
 9 WR      EQU    12
10 INDEX1  EQU    13
11 INDEX2  EQU    14
12 R15     EQU    15
13 DELAY   STM    12,RETURN             ENTRY POINT
14         LH     WR,2(R15)             FIND FORTRAN ADDRESS OF X
15         LH     INDEX2,0(WR)          LOAD X INTO INDEX2
16         BNP    STOP
17 LOOP2   LHI    INDEX1,100
18 LOOP1   SLLS   WR,15
19         SIS    INDEX1,1
20         BP     LOOP1
21         SIS    INDEX2,1
22         BP     LOOP2
23 STOP    LM     12,RETURN
24         AIS    R15,4
25         BR     R15
26 RETURN  DS     8
27         END
```

APPENDIX N

Listings for Program GEN1 and its Function Subprograms STUDIN, FISHIN, and FISH

This program is not called by the program REVERB. However, it is required to generate the t-distribution table read from logical unit 4 by the function subprogram STUDIM.

PROGRAM GEN1

```
$ASSM
GEN1 PROG GEN1 - PROGRAM TO GENERATE STORAGE FILE FROM STUDIN
$FORT
C   ------------------------------
C   THIS PROGRAM GENERATES A STORAGE FILE FROM THE SUBPROGRAM STUDIN,
C   WHICH GIVES THE INVERSE OF STUDENT'S T-DISTRIBUTION.  THIS FILE
C   IS GENERATED FOR ONE SPECIFIED VALUE OF ALPHA, AND CONSISTS OF THE
C   T-VALUES CORRESPONDING TO THE DEGREES OF FREEDOM FROM 1 TO 1000.
C
C   THE FILE IS INTENDED FOR USE BY THE SUBPROGRAM STUDIN, WHICH IS A
C   SUBSTITUTE FOR STUDIN FOR THOSE CASES WHEN THE MEMORY SPACE REQUIRED
C   BY STUDIN IS NOT AVAILABLE.
C
C   SUBPROGRAMS CALLED:  STUDIN.
C        [INDIRECTLY]:  FISHIN,FISH,TINORM
C   ------------------------------
      DIMENSION T(50)
      INTEGER*2 IT(50),FF
      DATA FF/X'0C00'/
      WRITE (5,10)
10    FORMAT (X/X/'THIS PROGRAM GENERATES A STORAGE FILE OF THE T-VALUES
     1 OF STUDENT''S'/'T-DISTRIBUTION FOR A SPECIFIED VALUE OF ALPHA.'/
     2X/A/'THE FOLLOWING LOGICAL UNIT ASSIGNMENTS ARE REQUIRED:'/X/
     35X,'LU1 -- STORAGE FILE    (22 SECTORS REQUIRED)'/
     45X,'LU2 -- SCRATCH FILE'/
     55X,'LU3 -- HIGH SPEED PRINTER'/
     65X,'LU5 -- CRT TERMINAL'/X/X/
     7'DO YOU WANT TO CHANGE THE LOGICAL UNIT ASSIGNMENTS?')
      READ (5,20) QS
20    FORMAT (A4)
      IF (QS.EQ.'NO') GO TO 30
      PAUSE 1
30    WRITE (5,35)
35    FORMAT ('DO YOU WANT A PRINTOUT OF THE FILE ONLY?')
      READ (5,20) QS
      IF (QS.NE.'NO') GO TO 140
      WRITE (5,40)
40    FORMAT ('ENTER ALPHA    (X.XX)'/
     1T7,'NOTE:  THIS VALUE IS 2 TIMES THE VALUE BEING USED IN TABLE
     2 5' T14. OF BOWKER & LIEBERMAN''S "ENGINEERING STATISTICS"')
      READ (5,50) ALPHA
50    FORMAT (F10.5)
      WRITE (5,60)
60    FORMAT ('ENTER MAX. NO. OF ITERATIONS   (3 DIGITS)')
      READ (5,70) ICMAX
70    FORMAT (I3)
      REWIND 1
      REWIND 2
      WRITE (1) ALPHA
      N=0
      DO 110 I=1,20
      DO 100 I=1,50
      N=N+1
      T(I)=STUDIN(ALPHA,N,ICMAX,IC)
      IT(I)=IC
      WRITE (5,80) N,T(I),IC
80    FORMAT ('N = ',I4,7X,'T = ',F8.3,6X,I3,' ITERATIONS')
100   CONTINUE
      WRITE (1) T
      WRITE (2) IT
```

```
110   CONTINUE
      ENDFILE 1
      WRITE (5,130)
130   FORMAT ('ENTER "CONTINUE" TO PRINT OUT TABLE')
      PAUSE 1
140   REWIND 1
      READ (1) ALPHA
      WRITE (3,150) ALPHA,(I,I=1,10)
150   FORMAT (X/X/X/T38,'T-DISTRIBUTION TABLE',10X,'ALPHA = ',F6.4/
     1X/X/X/5X,10(I2,9X)/X/X)
      N=0
      DO 170 J=1,20
      READ (1) T
      DO 170 K=1,5
      N=N+10
      K2=10*K
      K1=K2-9
      WRITE (3,160) (T(I),I=K1,K2),N
160   FORMAT (10(F10.5,X),4X,I4)
170   CONTINUE
      WRITE (3,180) FF
180   FORMAT (A1)
      REWIND 2
      WRITE (3,200) (I,I=1,10)
200   FORMAT (X/X/X/T50,'NO. OF ITERATIONS'/X/X/X/
     15X,10(I2,9X)/X/X)
      N=0
      DO 220 J=1,20
      READ (2) IT
      DO 220 K=1,5
      N=N+10
      K2=10*K
      K1=K2-9
      WRITE (3,210) (IT(I),I=K1,K2),N
210   FORMAT (4X,10(I3,8X),I4)
220   CONTINUE
      WRITE (3,180) FF
      END
```

FUNCTION SUBPROGRAM STUDIN

```
$ASSM
STUDIN PROG STUDIN - SUBPROG. GIVING INVERSE OF STUDENT'S T-DISTRIBUTION
$FORT
C    ------------------------------
C    THIS SUBPROGRAM COMPUTES THE VALUE OF THE INVERSE OF STUDENT'S
C    T-DISTRIBUTION.  IT IS A VERSION OF THE UNIVAC 1108 STAT-PACK
C    ROUTINE STUDIN.
C
C    NOTE--THE VALUE (T) COMPUTED BY THIS PROGRAM RELATES TO THE VALUES
C    GIVEN IN TABLE 3 OF THE APPENDIX OF BOWKER & LIEBERMAN'S
C    "ENGINEERING STATISTICS" AS FOLLOWS:
C    THE TABLE GIVES THE VALUES T(ALPHA:N)
C    THIS PROGRAM COMPUTES STUDIN(ALPHA1.N). WHERE STUDIN = T. WHICH
C    IS THE SAME VALUE GIVEN IN THE TABLE IF ALPHA1 = 2*ALPHA.
C    THUS. TO REPRODUCE THE VALUES OF TABLE 3. USE THE RELATION
C    T = STUDIN(2.*ALPHA.N,ICMAX.IC)
C    (THERE SEEMS TO BE SOME VARIATION IN THE DEFINITION OF ALPHA
C    AMONG DIFFERENT TEXTS.  THE STAT-PACK ROUTINE STUDIN CALLS
C    IT THE CONFIDENCE COEFFICIENT.)
C
C    FORTRAN CALLING SEQUENCE:
C  > VARIABLE = STUDIN(ALPHA,N,ICMAX,IC)
C
C    STUDIN (REAL) = THE CALCULATED VALUE OF T
C    ALPHA  (REAL) = THE CONFIDENCE COEFFICIENT
C                    (ALPHA MUST LIE BETWEEN 0.0 AND 1.0)
C    N      (INT)  = THE NO. OF DEGREES OF FREEDOM
C    ICMAX  (INT)  = MAXIMUM LIMIT FOR ITERATION COUNTER
C                    (10 SEEMS TO BE SUFFICIENT; THE ITERATION APPLIES
C                    TO FISHIN, WHICH IS CALLED BY STUDIN.)
C    IC     (INT)  = ITERATION COUNTER
C
C    OTHER SUBPROGRAMS CALLED:  FISHIN
C                  (INDIRECTLY):  TINORM. FISH
C    ------------------------------
      FUNCTION STUDIN(ALPHA,N,ICMAX,IC)
      BETA=1.-ALPHA
      STUDIN=1./SQRT(FISHIN(BETA,N,1.(ICMAX,IC))
      RETURN
      END
```

FUNCTION SUBPROGRAM FISHIN

```
$ASSM
FISHIN PROG FISHIN - SUBPROG. GIVING INVERSE OF FISHER'S F-DISTRIBUTION
$FORT
C   ------------------------------
C   THIS SUBPROGRAM COMPUTES THE VALUE OF THE INVERSE OF FISHER'S
C   F-DISTRIBUTION.  IT IS A VERSION OF THE UNIVAC 1108 STAT-PACK
C   ROUTINE FISHIN.
C
C   NOTE--THE VALUE (F) COMPUTED BY THIS PROGRAM RELATES TO THE VALUES
C   GIVEN IN TABLE 4 OF THE APPENDIX OF BOWKER & LIEBERMAN'S
C   "ENGINEERING STATISTICS" AS FOLLOWS:
C   THE TABLE GIVES THE VALUES F(ALPHA:N1.N2)
C   THIS PROGRAM COMPUTES FISHIN(ALPHA.N1.N2). WHERE FISHIN = F. THE
C   SAME VALUE GIVEN IN THE TABLE.  (ALPHA IS CALLED THE CONFIDENCE
C   COEFFICIENT.)
C
C   FORTRAN CALLING SEQUENCE:
C > VARIABLE = FISHIN(ALPHA.N1.N2.ICMAX.IC)
C
C   FISHIN (REAL) = THE CALCULATED VALUE OF F
C   ALPHA  (REAL) = THE CONFIDENCE COEFFICIENT.
C                   (ALPHA MUST LIE BETWEEN 0.0 AND 1.0)
C   N1     (INT)  = THE DEGREES OF FREEDOM OF THE FIRST SAMPLE
C   N2     (INT)  = THE DEGREES OF FREEDOM OF THE SECOND SAMPLE
C   ICMAX  (INT)  = MAXIMUM LIMIT FOR ITERATION COUNTER
C                   (10 SEEMS TO BE SUFFICIENT: MOST OF THE VALUES IN
C                   THE COLUMN FOR N1=1 DO NOT CONVERGE TO THE CHOSEN
C                   LIMITS EVEN AFTER 100 ITERATIONS.  HOWEVER, NO
C                   GAIN IN THE ACCURACY IS OBSERVED AFTER THE FIRST
C                   FEW ITERATIONS.  100 ITERATIONS REQUIRE AT LEAST
C                   20 SECONDS.  FOR ALL VALUES FOR N1 AND N2 GREATER
C                   THAN 1. CONVERGENCE SEEMS TO OCCUR IN LESS THAN
C                   10 ITERATIONS. USUALLY 3.)
C   IC     (INT)  = ITERATION COUNTER
C
C   OTHER SUBPROGRAMS CALLED:  TINORM, FISH
C   -------------------------  -------
      FUNCTION FISHIN(ALPHA.N1.N2.ICMAX.IC)
      DOUBLE PRECISION FSHNDP.Y1.Y2.X.Y.H.G.GLOG.C
      Y1=N1
      Y2=N2
C     ---------------------------------------
C     ADJUST FOR DEGREES OF FREEDOM EQUAL TO 1
C     ---------------------------------------
      IF (N1.EQ.1) Y1=2.
      IF (N2.EQ.1) Y2=2.
C     ------------------------------------------------------
C     CALL TINORM TO GET INVERSE NORMAL VALUE OF 1.-ALPHA
C     ------------------------------------------------------
      X=TINORM(1.-ALPHA)
C     ------------------
C     COMPUTE LAMBDA VALUE
C     ------------------
      Y=(X*X-3.)/6.
      IC=1
C     ------------------------------------------------------------
C     COMPUTE INITIAL APPROXIMATION TO THE INVERSE 'F' FUNCTION
C     ------------------------------------------------------------
      Y1=1./(Y1-1.)
      Y2=1./(Y2-1.)
```

```
      H=2./(Y1+Y2)
      X=X*DSQRT(H+Y)/H-(Y1-Y2)*(Y+5./6.-2./(3.*H))
      X=DEXP(2.*X)
C     ------------------------------------------
C     COMPUTE THE CONSTANT TO THE 'F' DISTRIBUTION.
C     TESTING FOR N1 AND/OR N2 ODD OR EVEN
C     ------------------------------------------
      G=1.
      IB1=2
      IF (MOD(N1,2).EQ.0) GO TO 10
      G=1.7724539
      IB1=1
   10 IB2=2
      IF (MOD(N2,2).EQ.0) GO TO 20
      G=G*1.7724539
      IB2=1
   20 IB3=2
      IF (MOD(N1+N2,2).EQ.0) GO TO 30
      G=G/1.7724539
      IB3=1
   30 IF ((IB1+IB2).NE.2) G=2.*G
      GLOG=DLOG10(G)
      IF ((N1+N2).LE.3) GO TO 50
      ND=N1+N2-2-IB3+1
      DO 40 I=1,ND,2
      IF ((IB1+I-1).LE.(N1-2)) GLOG=GLOG+ALOG10(IB1+I-1.)
      IF ((IB2+I-1).LE.(N2-2)) GLOG=GLOG+ALOG10(IB2+I-1.)
   40 GLOG=GLOG-ALOG10(IB3+I-1.)
C     ------------------------------------------------------------
C     COMPUTE THE VALUE OF FISHIN (DOUBLE PRECISION VALUE = FSHNDP)
C     ------------------------------------------------------------
   50 Y2=N2/(N2+N1*X)
      Y1=1.-Y2
      XSP=X
      K=1
      Y=1.-ALPHA-FISH(XSP,N1,N2)
      IF (Y.LT.0.) K=-1
      IF (Y.EQ.0.) GO TO 60
      Y=DLOG10(DABS(Y))
      C=(N1*DLOG10(Y1)+N2*DLOG10(Y2))/2.
      Y=GLOG+Y-C
      IF (Y.LT.-75.) GO TO 55
      Y=10.**Y
      GO TO 60
   55 Y=0.
   60 Y=1.+K*Y
      FSHNDP=X/Y
C     ------------------------------------------------------------
C     IF FISHIN (FSHNDP) IS NEGATIVE, RESET TO .5*LAST APPROXIMATION(X)
C     ------------------------------------------------------------
      IF (Y.LT.0.) FSHNDP=.5*X
C     ------------------------------------------------------
C     IF ABS. VALUE OF THE DIFFERENCE IS LESS THAN .5D-6, RETURN
C     ------------------------------------------------------
      IF (DABS(X/FSHNDP-1.).LT.(.5D-6)) GO TO 70
C     ------------------------------------------------------
C     IF REL. VALUE OF THE DIFFERENCE IS LESS THAN .5D-6, RETURN
C     ------------------------------------------------------
      IF (DABS(X-FSHNDP).LT.(.5D-6)) GO TO 70
      IF (IC.GE.ICMAX) GO TO 70
```

```
121         IC=IC+1
122 C     ------------------------------------------------------------
123 C     SET APPROXIMATION EQUAL TO FISHIN (FSHNDP). CONTINUE TO ITERATE
124 C     ------------------------------------------------------------
125         X=FSHNDP
126         GO TO 50
127    70   FISHIN=FSHNDP
128         RETURN
129         END
```

FUNCTION SUBPROGRAM FISH

```
$ASSM
FISH PROG FISH - FCTN SUBPROGRAM FOR EVALUATING FISHER'S F-DISTRIBUTION
$FORT
C   ------------------------------
C   THIS SUBPROGRAM COMPUTES THE VALUE OF FISHER'S F-DISTRIBUTION
C   AT THE POINT F WITH N1.N2 DEGREES OF FREEDOM.  IT IS A VERSION
C   OF THE UNIVAC 1108 STAT-PACK ROUTINE FISH.
C
C   THE VALUE OF FISHER'S DISTRIBUTION = THE PROBABILITY THAT A VARIABLE
C   DISTRIBUTED WITH FISHER'S F-DISTRUBUTION WITH N1.N2 DEGREES OF
C   FREEDOM IS LESS THAN OR EQUAL TO A GIVEN VALUE (F).
C
C   NOTE--THIS PROBABILITY RELATES TO THE VALUES GIVEN IN TABLE 4 OF
C   THE APPENDIX OF BOWKER & LIEBERMAN'S "ENGINEERING STATISTICS"
C   AS FOLLOWS:
C   THE TABLE GIVES THE VALUES F(ALPHA:N1.N2)
C   THIS PROGRAM COMPUTES FISH(F.N1.N2) WHERE
C   FISH = BETA = 1.-ALPHA
C   BETA = PROBABILITY [ F(ALPHA:N1.N2) < OR = F ]
C   (ALPHA IS CALLED THE CONFIDENCE COEFFICIENT IN THE STAT-PACK
C   ROUTINE FISHIN.)
C
C   FORTRAN CALLING SEQUENCE:
C > VARIABLE = FISH(F.N1.N2)
C
C   FISH   (REAL) = THE CALCULATED VALUE OF THE PROBABILITY
C   F      (REAL) = THE VALUE AT WHICH THE PROBABILITY IS CALCULATED
C                   (F CANNOT BE LESS THAN 0.0)
C   N1     (INT)  = THE DEGREES OF FREEDOM OF THE FIRST SAMPLE
C   N2     (INT)  = THE DEGREES OF FREEDOM OF THE SECOND SAMPLE
C
C   NO OTHER SUBPROGRAMS CALLED
C   ------------------------------
      FUNCTION FISH(F.N1.N2)
      DOUBLE PRECISION FISHDP.W.Y.H.C.HLOG
      LOGICAL E1.E2.E3
C     ---------------------- -------------------------
C     INITIALIZATION AND SETTING OF LOGICAL SWITCHES
C     TO .TRUE. IF THE DEGREES OF FREEDOM ARE EVEN
C     ---------------------- ---- ---------------------
      E1=.FALSE
      E2=.FALSE.
      E3=.FALSE.
      IF (MOD(N1.2).EQ.0) E1=.TRUE.
      IF (MOD(N2.2).EQ.0) E2=.TRUE.
      X=N2/(N2+N1*F)
      IF (.NOT.(E1.OR.E2)) GO TO 50
      IF (E1.AND..NOT.E2) GO TO 20
      IF (.NOT.E1.AND.E2) GO TO 10
      IF (N1.LE.N2) GO TO 20
C     ---------- ---- ------- ---- ----------------------
C     INITIALIZATION FOR SECOND DEGREE OF FREEDOM EVEN AND
C     LESS THAN FIRST DEGREE OF FREEDOM IF IT, TOO, IS EVEN
C     ---- ------------------------------------------------
   10 I=N1
      N1=N2
      N2=I
      X=1.-X
      E3=.TRUE.
C     ---------- ---- ------- ---- ----------------------
```

```
C     INITIALIZATION FOR FIRST DEGREE OF FREEDOM EVEN AND
C     LESS THAN SECOND DEGREE OF FREEDOM IF IT, TOO, IS EVEN
C     ------------------------------------------------------
   20  Y=1.-X
C     ------------------------------------------------------------
C     COMPUTE PROBABILITY FOR AT LEAST ONE DEGREE OF FREEDOM EVEN
C     ------------------------------------------------------------
       FISHDP=0.
       HLOG=(N2/2.)*DLOG10(X)
       I1=1
   22  IF (HLOG.GT.-75.) GO TO 28
       HLOG=HLOG+DLOG10(Y*(N2+2.*(I1-1))/(2.*I1))
       I1=I1+1
       GO TO 22
   28  H=10.**HLOG
       M=N1/2.
       IF (I1.GT.M) GO TO 32
       DO 30 I=I1,M
       FISHDP=FISHDP+H
   30  H=(H*Y*(N2+2.*(I-1)))/(2.*I)
   32  FISH=FISHDP
       IF (E3) GO TO 40
C     ----------------------------------------
C     ADJUST CALCULATED PROBABILITY IF ITS
C     ONES COMPLEMENT WAS CALCULATED ORIGINALLY
C     ----------------------------------------
       FISH=1.-FISH
       RETURN
   40  I=N1
       N1=N2
       N2=I
       RETURN
C     ------------------------------------------------------
C     COMPUTE PROBABILITY FOR BOTH DEGREES OF FREEDOM ODD
C     ------------------------------------------------------
   50  Y=1.-X
       H=.6366197723*SQRT(X*Y)
       C=DSQRT((1./X)-1.)
       FISHDP=.6366197723*DATAN(C)
       IF (N2.EQ.1) GO TO 70
       M=N2-2
       DO 60 I=1,M,2
       FISHDP=FISHDP+H
       H=H*X*(I+1)/(I+2)
       IF (H.LT.1.D-75) GO TO 62
   60  CONTINUE
       GO TO 70
   62  IF (I.EQ.M) GO TO 70
       I1=I+2
       HLOG=DLOG10(H)
       DO 64 I=I1,M,2
   64  HLOG=HLOG+DLOG10(X*(I+1)/(I+2))
       GO TO 72
   70  HLOG=DLOG10(H)
   72  IF (N1.EQ.1) GO TO 90
       C=N2
       HLOG=HLOG+DLOG10(C)
       I1=1
   74  IF (HLOG.GT.-75.) GO TO 78
       HLOG=HLOG+DLOG10(Y*(N2+I1)/(I1+2))
```

```
121         I1=I1+2
122         GO TO 74
123    78   H=10.**HLOG
124         M=N1-2
125         IF (I1.GT.M) GO TO 90
126         DO 80 I=I1,M,2
127         FISHDP=FISHDP-H
128    80   H=H*Y*(N2+I)/(I+2)
129    90   FISH=FISHDP
130         RETURN
131         END
```

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date: Yr. 77 Mo. 10 Day 28

02. Summary prepared by (Name and Phone): Thomas W. Bartel (301) 921-3783

03. Summary action: New [X] Replacement [ ] Deletion [ ]
Previous Internal Software ID:

04. Software date: Yr. 76 Mo. 12 Day 30

05. Software title: Interactive Computer Program for the Determination of Reverberation Time

06. Short title: REVERB

07. Internal Software ID: None

08. Software type:
- [ ] Automated Data System
- [X] Computer Program
- [ ] Subroutine/Module

09. Processing mode:
- [X] Interactive
- [ ] Batch
- [ ] Combination

10. Application area — General:
- [ ] Computer Systems Support/Utility
- [X] Scientific/Engineering
- [ ] Bibliographic/Textual
- [ ] Management/Business
- [ ] Process Control
- [ ] Other

Specific: Acoustics; reverberation room measurements

11. Submitting organization and address:

Institute for Basic Standards
National Bureau of Standards
Washington, D.C. 20234

12. Technical contact(s) and phone:

Thomas W. Bartel
(301) 921-3783

13. Narrative

The program is used to measure the reverberation time in a reverberation room according to ASTM C423-66, Standard Method of Test for Sound Absorption of Acoustical Materials in Reverberation Rooms. The computer controls several elements of the instrumentation system through a special interface, acquires the digitized sound pressure levels of the decaying reverberant sound field, and determines the reverberation time through a linear least-squares computation. Transmission of data to and from the instrumentation system is handled through separate subroutines written in the computer's assembly language.

14. Keywords

Acoustics; computer-controlled instrumentation system; real-time analyzer; reverberation room; reverberation time.

15. Computer manuf'r and model	16. Computer operating system	17. Programing language(s)	18. Number of source program statements
Interdata Model 70	RTOS	Fortran V, Level 1, Interdata Assembly lang.	1436

19. Computer memory requirements	20. Tape drives	21. Disk/Drum units	22. Terminals
35,452 8-bit bytes	None	One/2.5 megabyte unit	One

23. Other operational requirements

132-character line printer
real-time analyzer, random noise generator, microphone multiplexer interfaced to computer

24. Software availability:
- Available [X]
- Limited [ ]
- In-house only [ ]

Paper or mag tape

25. Documentation availability:
- Available [X]
- Inadequate [ ]
- In-house only [ ]

Internal report

26. FOR SUBMITTING ORGANIZATION USE

185-101

Standard Form 185
1974 July
U.S. Dept. of Commerce—NBS
(FIPS. Pub. 30)

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 77-1383	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

Interactive Computer Program for the Determination of Reverberation Time

5. Publication Date: December 1977

6. Performing Organization Code: 200.03

7. AUTHOR(S)

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No.: 2003116

11. Contract/Grant No.

12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP)

Same

13. Type of Report & Period Covered: Final

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)

A description of the computer program used to measure the reverberation time in a reverberation room is presented. The program controls the operation of a real-time analyzer, a random noise generator, and a microphone multiplexer. The reverberation time for each digitally recorded decay curve is determined from a straight line least-squares fit. The program is written in FORTRAN V and requires approximately 35,000 eight-bit bytes of core memory. Flow charts, source listings, and sample printouts are included.

17. KEY WORDS (six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)

Automated data acquisition; computer; reverberation room, reverberation time; software.

18. AVAILABILITY

[X] Unlimited

[ ] For Official Distribution. Do Not Release to NTIS

[ ] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13

[X] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
UNCLASSIFIED	126
20. SECURITY CLASS (THIS PAGE)	22. Price
UNCLASSIFIED	$7.25

USCOMM-DC 29042-P74